

**SJ Research**  
**Modular Disc File Server Manual**  
**Release 1.00 - 29th April 1987**



## Copyright

©SJ Research Ltd.  
108 Mill Road  
Cambridge  
CB1 2BD

Tel. Cambridge (0223) 69927

Whilst copyright in this document is reserved by SJ Research Ltd, permission is hereby given to owners of SJ Research File Servers to make copies of sections of this manual for their own use, and for the use of their pupils or colleagues.

The sale of such copies is expressly forbidden without explicit permission in writing from SJ Research Ltd.

## Terminology

To avoid cumbersome sentence construction, we have sometimes referred to users (and system managers) as 'he'. We wish to apologise to users and system managers who do not wish to be 'he', and hope that they will tolerate some terminological inaccuracy in the interests of readability.

## Typeface Usage

Throughout this manual, `Courier` is used for machine-generated output and **Courier** is used for user-typed input. Other typefaces are used for emphasis, or for defining new terms.

All goods and services provided by SJ Research Limited are subject to the company's standard Terms and Conditions of Sale. A copy of such terms is available on request and such standard terms shall apply on and from the date of actual commencement of such service or the provision of goods to which the same relate.



# Contents

## Section A - Introduction

### 1. Introduction

## Section B - For Users

### 2. Introduction to File Server facilities

- 2.1 What is a Network?
- 2.2 Logging on and simple filing operations
  - 2.2.1 Starting up the Econet system
  - 2.2.2 Simple filing operations
  - 2.2.3 Common error messages with Econet
- 2.3 The Econet filing system
  - 2.3.1 Hierarchical directories
  - 2.3.2 Selecting directories and using pathnames
- 2.4 Accounts and access
  - 2.4.1 The accounting system and getting information about files
  - 2.4.2 Controlling access to files
- 2.5 Stations and printing through the network
  - 2.5.1 Different types of station
  - 2.5.2 Selecting a network printer
  - 2.5.3 Turning the printer on and off
  - 2.5.4 Direct printing of files from the File Server
  - 2.5.5 Banners and logical printers
  - 2.5.6 The print queue directory
- 2.6 Other facilities available on the network
  - 2.6.1 Passwords, Libraries and boot files
  - 2.6.2 General information available on the network
  - 2.6.3 Copying files and directory structures
  - 2.6.4 Communicating with other network users

### 3. Utility programs reference

- 3.1 Introduction
  - 3.1.1 Syntax definitions
  - 3.1.2 File, directory and general specifiers
  - 3.1.3 Wild card specifiers
  - 3.1.4 Disc names
- 3.2 Summary of Commands
- 3.3 Command Details

## Section C - For System Managers

### 4. An Introduction to system management

- 4.1 Operating instructions for the MDFS
  - 4.1.1 Switching on the File Server
  - 4.1.2 Switching off the File Server
  - 4.1.3 Changing discs
- 4.2 Organising the File Server
- 4.3 Command Examples
- 4.4 The File Server program
  - 4.4.1 Copying new File Server software onto a winchester disc

### 5. Setting up users

- 5.1 Overview
  - 5.1.1 Keeping a list of users
  - 5.1.2 Entering users onto the system
- 5.2 Listusers
- 5.3 Editpass
  - 5.3.1 Using Editpass
  - 5.3.2 Keeping a security copy of the password file
  - 5.3.3 Password files on more than one disc
- 5.4 Finishing the process
  - 5.4.1 Creating user root directory
  - 5.4.2 Setting the account numbers of user root directories
  - 5.4.3 Distinction between users' names and user root directories
  - 5.4.4 Crediting or debiting an account

## 6. Setting up printers

- 6.1 Overview
- 6.2 Everyday printer management
- 6.3 Configuring the printers
  - 6.3.1 Editprint
  - 6.3.2 The banner file
    - 6.3.2.1 Table of standard characters
    - 6.3.2.2 Creating the banner file
    - 6.3.2.3 Example
- 6.4 The remote despooler

## 7. Utility Mode

- 7.1 Introduction
- 7.2 Entering Utility Mode on the MDFS
  - 7.2.1 For users with a FAST ROM fitted in a BBC Microcomputer
  - 7.2.2 For users with no FAST ROM
- 7.3 Using Utility Mode
  - 7.3.1 General notes
  - 7.3.2 Alter parameters
  - 7.3.3 Boot File Server
  - 7.3.4 Whole disc copy
  - 7.3.5 dd defect list
  - 7.3.6 Format new disc
  - 7.3.7 List discs
  - 7.3.8 Clear password file
  - 7.3.9 Rename disc
  - 7.3.10 Set station number
  - 7.3.11 Tape menu
  - 7.3.12 Verify disc
  - 7.3.13 Park disc heads
- 7.4 Resetting the Station Number to 254
- 7.5 Drive Letter Allocation
  - 7.5.1 Physical drives, logical discs and partitions
  - 7.5.2 Winchester disc lettering

## 8. Using the Tape

- 8.1 Overview
  - 8.1.1 The tape drive
  - 8.1.2 Inserting a cartridge and autoloader
  - 8.1.3 Removing the cartridge and unloading
  - 8.1.4 Tape cartridges
  - 8.1.5 Tape cartridges life-expectancy
  - 8.1.6 Managing tape backups
- 8.2 Off-line operation
  - 8.2.1 Using the backup command
  - 8.2.2 Formatting tapes
  - 8.2.3 Getting tape information

- 8.2.4 Restoring data from a tape backup
- 8.2.5 Other operations
- 8.3 On-line operation
- 8.3.1 Example file recovery

## 9. Dealing with problems

- 9.1 Overview
- 9.2 What to do when the File Server crashes
  - 9.2.1 Mains electricity
- 9.3 Debugging the network hardware
  - 9.3.1 Accidents
  - 9.3.2 Recovering from mistakes
  - 9.3.3 Deliberate interference
  - 9.3.4 Special techniques
- 9.4 Debugging the network hardware

## Section D - For Programmers

### 10. Programmers Reference

- 10.1 OSGBP
- 10.2 OSFIND
- 10.3 OSBGET
- 10.4 OSBPUT
- 10.5 OSARGS
- 10.6 OSFILE
- 10.7 OSCLI
- 10.8 OSWORD
- 10.9 Transmit calls
- 10.10 Receive calls
- 10.11 Reading arguments call
- 10.12 Miscellaneous station environment calls
- 10.13 Send to File Server
- 10.14 OSBYTE
- 10.15 Transmitting
- 10.16 Receiving
- 10.17 Port numbers
- 10.18 The bridge
- 10.19 Printers
- 10.20 File Server Interface
- 10.21 Password file format
- 10.22 Application notes
- 10.23 The four-way handshake and NETMON

## Appendices

### Appendix A - Error messages given by BBC Computers

- A.1 Network related error messages generated locally
- A.2 File Server errors reported as Error A8
- A.3 File Server errors with correct error numbers

### Appendix B - System Errors

- B.1 File Server Messages
  - B.1.1 Disc and Tape error examples
- B.2 Hard-disc errors
  - B.2.1 Non-fatal errors
  - B.2.2 Errors accompanied by the system error LED
  - B.2.3 Flashing System error LED after powering on an MDFS
- B.3 Tape errors
  - B.3.1 Rodime disc error codes
  - B.3.2 Adaptec ACB 4000A/4070 disc error codes

- B.4 SCSI sense key definitions
- B.5 Tape errors
- B.6 Floppy disc errors

#### Appendix C - Installing your system

- C.1 Installing the File Server
  - C.1.1 Indicator lights
- C.2 Installing Printers
- C.3 Configuring SCSI peripherals
  - C.3.1 General notes
  - C.3.2 Setting the controller number and terminators
  - C.3.3 Installing tape drives
  - C.3.4 Installing SJ Research winchesters
  - C.3.5 Installing BBC compatible winchesters

# Chapter 1: Introduction to the SJ Research File Server

---

## 1.0 How to use this Manual

This manual is split into sections covering different aspects of operation of the File Server, as follows

<b>Section A</b>	<b>Introduction</b>	Chapter 1
<b>Section B</b>	<b>For Users</b>	Chapter 2, Chapter 3
<b>Section C</b>	<b>For System Managers</b>	Chapter 4, Chapter 5, Chapter 6, Chapter 7, Chapter 8, Chapter 9
<b>Section D</b>	<b>For Programmers</b>	Chapter 10
<b>Appendices</b>	<b>A Error Messages given by BBC Computers</b> <b>B System Errors</b> <b>C Installing Your System</b>	

The rest of this chapter gives an overall description of the SJ Research File Server system.

## 1.1 Introduction to the SJ Research File Server Systems

The SJ Research Modular Disc File Server is designed to work on Econet® local area computer networks, to the specification laid down by Acorn Computers Ltd as "Econet Level 2" but with a number of additional features. The full features of the SJ Research systems are briefly described below:

- \* Fully *random access* to files is offered, either one byte at a time, or a specified block of data (up to the entire file) at a time.
- \* The concept of *fileowners* (with passwords if required) is offered to restrict access to files by unauthorised users.
- \* Fileowners' directories are themselves files (of a special sort) so that a fileowner can have sub-directories within other directories. In addition, the *root* directory of the system shall be accessible to users, allowing them to access (where authorised) files belonging to other fileowners.
- \* The Modular Disc File Server allows four floppy disc drives and up to two hard discs (up to 140 megabytes). The MD FS stores data, on the floppy disc drives, using double density, which means that on each standard 80 track disc drive 800 kilobytes (K) of data can be stored.
- \* In addition, the MD FS supports a tape streamer for backing up hard discs. This allows important data to be protected, off-site if need be; data can also be interchanged on this medium.
- \* As the MD FS can use both hard discs and floppy discs this allows the hard disc to be used for general purpose software whilst the floppy disc can contain software specific to a particular group of people.
- \* A system of *accounts* is provided, to control both the use of disc space by users, and owner access to files. Each fileowner is allocated one or more accounts, and each account is given a *balance* by the person in charge of the system (the System Manager). Account numbers range between 0 and FF (hexadecimal), although the system manager could of course use 0 to 99 if this is simpler. By allocation of a suitable balance to each account, users can be encouraged to remove unwanted files.

- \* The account number given to a file controls the *owner access* to the file, that is to say, the user with access to the account number of a file may read, write or delete that file; if the file is a directory, he may create new files in that directory.
- \* Each file has also an *auxiliary account number*. This can be set by the fileowner. The user(s) with access to the account with number equal to the file's auxiliary account number, also has owner access to that file.
- \* If an attempt is made to load or open any file that is not in the currently selected directory, the *library directory* will be searched for the file. This will occur regardless of the method of access to the file (Level 2 specifies this feature only for \* commands). When a user logs-on, the file server will **automatically** select the directory LIBRARY (if it exists) as library directory.
- \* A *real-time clock* is included in the File Server. When the full information about a file is requested, the system will display the date of first creation of the file, and the date and time that the file was last updated. In addition, utility programs (for example \*TIME) are provided to read the clock directly. The clock runs from an internal rechargeable battery when the system is switched off.
- \* A *printer server* is included in the unit, allowing any station on the network to route its output to a printer, via the network itself.
- \* The MDFS supports *printer spooling* so that output routed to a busy printer will be stored on disc until the printer becomes free. The print queue is accessible to users, who can thus control the order in which jobs are printed when they have access to appropriate account numbers.

Note that the additional feature of accounts need not affect the ordinary users, unless they specifically wish to make use of them. In fact, the system manager may turn off the accounting system completely where no data security is required: in this case all users would have access to everything on the system. Alternatively, he may keep data security but disable the disc space accounting part of the system, by setting all accounts to a suitably large value. See Section 3.2 (under \*ACCOUNT and \*ACCESS) for a full description of account control.

## Chapter 2: Introduction to File Server Facilities

---

This chapter gives an introductory guide to use of the Econet® system on the BBC Microcomputer. Section 2.1 explains what a network is, and what it can do. The following sections then give simple instructions for logging on and using the network for filing operations, and go on to discuss the directory structure and accounting facilities provided on the File Server. The chapter concludes with explanations of printing and the other facilities available on the network.

Chapter 3.3 gives a full description of each filing system command, and also of some of the utility programs available to make your life easier. A complete reference list of Econet error messages is given in Appendix A.

Advanced programmers, and especially machine code programmers will want to refer to Section D.

### 2.1 What is a Network ?

A computer network is a method of connecting a number of computers together, so that they can communicate with each other. There are several considerable advantages in doing this:

- \* Users can *communicate* with each other, and pass messages.
- \* Expensive equipment like printers can be put on the network, so that all users have access to them.
- \* A *network filing system* can be set up, using a central piece of equipment called a File Server. Again, this is cheaper than having a separate disc drive for each computer, but also it allows --
- \* File sharing, where many users can use the same program without needing separate copies. It is possible to have files which can be updated by several people independently, and other files which can be looked at but not changed by others.

The Econet network is designed to allow all these facilities on the BBC Microcomputer (and in fact on a range of other machines). The special electronic circuits to communicate with the network are built into the computer, and a special piece of software, the *Network Filing System*, is fitted in a read-only memory chip (ROM) to control the network electronics.

All computers have one or more filing systems built into them, if they are to be of much use. A filing system allows you to store away programs and data when you have finished working with them, so that you can use them another day without having to type everything in again from the keyboard.

The most common filing system is the tape cassette system, built into almost every microcomputer when you buy it. It allows you to keep *files*: that is to say collections of information whether they are programs, data, text or anything else that you wish to keep for another day. The tape cassette filing system usually allows you to give a *name* to a file, so that, when you want that piece of information back, you can tell the computer the name, and it will search the tape until it finds the corresponding name, and then proceed to retrieve that file.

A useful feature of a filing system is that you can use it to produce a *catalogue* of all the files. On the BBC Micro, this is done by typing \*CAT. If you are using the tape cassette system, you will then have to start the tape, and the computer will print out each name as it comes to it.

There are two other filing systems which are also very common, the *disc* system, and the *network* system. Both require some extra circuitry and software in the computer. The disc system stores files on a *floppy disc*, which is a circular piece of plastic covered with magnetic coating (the same as on magnetic tape) in a cardboard sleeve. The floppy disc is inserted into a special drive, which rotates the plastic disc, and reads or

writes information through a special pickup head which can be moved radially to cover different areas of the disc.

The Econet network system uses a central computer called a *File Server*: this is a computer equipped with sufficient mass storage (usually floppy discs or larger hard discs) for the needs of all the users of the system. Other microcomputers, called *Client Stations* are connected to this File Server with cables, along which data is passed. The central machine runs programs to pass files back and forth when required, and keep track of all the files on the discs and who owns them. There is no restriction on the number of File Servers in a network.

## 2.2 Logging On and Simple Filing Operations

### 2.2.1 Starting Up the Econet System

Switch on the BBC Microcomputer and the monitor, and the following message should be displayed, if your station is connected to the network.

```
BBC Computer 32K
Econet Station nnn
BASIC
>
```

nnn is the number of your station on the network. If the message does not mention Econet (saying perhaps "Acorn DFS" if you have a disc interface as well), then hold down the letter N on the keyboard, and press and release the <Break> key.

If your default language is not BASIC, the third line of the message will be different (saying perhaps "VIEW" or "WORDWISE"). To change the language to BASIC, type:

**\*BASIC <Return>**

and press and release the <Break> key. The message above should now appear.

The second line of the message may read:

```
Econet Station nnn No Clock
```

in which case there is a minor problem with the network. See Section 2.2.3 for what to do.

If you want to select the Econet filing system without pressing <Break> (for example when you are in the middle of a program or other work), then use the command:

**\*NET**

This command may be used as part of a BASIC program, but it should be the last command in a program line (see under OSCLI in Section 3.2. for further details). Similarly, to use the disc or tape filing systems, type:

**\*DISC**            (or)  
**\*TAPE**

Having selected the Econet filing system, you will now have to identify yourself to the File Server, called logging on; so that when it is asked about one of your files it knows where in its storage to look. You will not be able to use the network filing system for anything until you have done this -- all you will get is the message:

Who are you?

The person in charge of the network -- the official title is System Manager -- will have given you a *User Identifier* and perhaps a password as well. Normally your User Identifier will be your name or your initials, and your password will be some random characters.

The usual method of logging on is to type the command:

**\*I AM <user identifier> [<password>]**

If you have not been given a password you should just type <Return> after the User Identifier; otherwise leave a space between your User Id and your password and type <Return> at the end. For example:

**\*I AM DIANA <Return> (or)**  
**\*I AM TONY WOMBAT47 <Return>**

The computer should reply with a > after a short pause. The prompt may be a different character if you are not in BASIC. If there is a pause followed by a message, then there is a problem -- see Section 2.2.3 for an explanation.

The system manager can set up an alternative method of logging on, which performs **\*I AM** automatically and may disable the normal command. If this is the case, after turning your station on you should hold the <Shift> key down, press and release the <Break> key, and then release the <Shift> key. This will cause the screen to clear and you will be prompted for your User Id, and password if necessary. The process might go, for example:

User Id :**JOHN <Return>**  
Password:**KITTEN <Return>**

This method should work in exactly the same way as **\*I AM**. Note that if the system is not set up to do automatic logging on, typing <Shift-Break> may change your filing system, depending on the defaults set up by the system manager.

If the network you are using has more than one File Server, or if the station number of the File Server is not 254, as is assumed by the BBC Microcomputer, you may need to type the File Server station number before your User Id. For example, if the File Server station number has been changed to 235:

**\*I AM 235 DIANA HUNGRY <Return>**

The system manager should tell you if you need to include a File Server number. If your network contains *bridges* onto other networks, these other networks are identified by a network number. To reach a File Server on one of these networks, you will need to give the *full station number* of the File Server, which will be of the form <network number>.<station number>. For example, to log on to the File Server at station 200 on network 5, type:

**\*I AM 5.200 MARY**

If the memory of your BBC Microcomputer has become corrupted, the default network number or File Server station number may have been changed to an inappropriate value. In that case you will need to re-specify the full station number before you will be able to log on. Note that the local network for your station is always referred to as network number 0, and not by its global number. Thus:

**\*I AM 0.254 STEPHEN 123GO**

will restore the selections to your local network and File Server number 254.

When you want to finish your session at a particular station, log off by typing the command:

**\*BYE <Return>**

If you do not log off in this way, other people will be able to use your file space and interfere with your files,

using the station at which you are logged on. It is however possible for you to be logged on to the same File Server from more than one station on the network, if you wish.

Further details of the **\*I AM** and **\*BYE** commands are given in Section 3.3

### 2.2.2 Simple filing operations

After you have logged on, you will be in your own personal area of the File Server storage space. This area will contain a list of the programs stored within it -- this is called a *directory*. To obtain a *catalogue* of this directory, type the command:

**\*CAT**

and you will be given a display of directory information, explained in Section 3.3 under **\*CAT**, of the form:

```
DIANA      (000)   Owner
MAIN-IV    Option 00 (Off)
Dir.DIANA  Lib.LIBRARY
```

It is likely that no files will be listed after this information, as you will not have stored any programs on the network filing system yet. We will do that now, by typing in the following BASIC program and saving it:

```
10 REM Hello there!
```

```
SAVE "TEST"
```

Now typing **\*CAT** will list the program name after the directory information, as:

```
TEST      WR/r
```

The letters after the program name refer to the access rights of the file and are discussed in Section 2.4. From this directory, typing:

```
LOAD "TEST"
```

will put this program back into the memory of the BBC computer. The command **CHAIN** will both load and run a program. When you have finished with the program, typing:

```
*DELETE TEST
```

will erase it from the File Server disc and remove its name from the directory. The name of an existing program can be changed, for example:

```
*RENAME TEST MESSAGE
```

will change the filename of a program from **TEST** to **MESSAGE**.

If, when using **SAVE**, you specify a filename that already exists in your directory, the new program will overwrite the old one. You can protect files against being overwritten (see Section 2.4., in which case the letter **L** will appear in the access letters for the file. Such a file cannot be deleted, renamed or overwritten until its protection has been removed.

If you press **<Break>** or **<Ctrl-Break>**, the current program will be cleared from your station's memory. The program can be recovered by using the BASIC command **OLD** immediately. If you **SAVE** the program before recovering it, an empty file will be created, overwriting any previous unprotected file of that name. It is thus a good idea to check that a program is there, perhaps by listing part of it, before saving it. The system manager may also have set up an option to prevent such short **SAVES**.

Note that **SAVE**, **LOAD** and **CHAIN** are BASIC commands and the filenames specified for them must be enclosed in quotation marks. Filing system commands, which always start with the character **\***, do not in

general need quotes around filenames, although they can be put in if you like.

The commands **\*SAVE**, **\*LOAD** and **\*RUN** are commands used with machine code programs and are described in Section 3.2..

### 2.2.3 Common Error Messages with Econet

#### The "No Clock" Message

The most likely cause of this message is that the your station is not plugged into the network. Check that the Econet plug in the back of the computer is plugged in firmly, and that the other end of the lead is plugged into the wall socket or cable adaptor that connects to the network. Then press N and <Break> together and see if the "No Clock" message goes away.

If it does not, tell the system manager, and try to log on at a different station. If you are the system manager, look up the "No Clock" error in Section A.1 and read Section 9.4.

#### Fault messages after **\*I AM** or other Filing Commands

There are several likely messages, including "Not listening", "Line jammed" and "Net error". A full explanation of all error messages is given in Appendix A.

The "Not listening" message happens when the station to which you are trying to talk is either non-existent, or switched off, or busy. This may occur if the File Server station number used in your log on command, or assumed by the BBC Microcomputer, does not correspond to a File Server on the network. Check the station number, network number and current status of the File Server to which you wish to log on. See Section 3.3. under **\*I AM** for more details.

The "Line jammed" or "Net error" messages may be cured by pressing <Ctrl-Break> or turning the power to your station off and on again, and then attempting to log on again. If this does not work, inform the system manager (who should refer to Chapter 9) and try logging on at another station.

## 2.3.The Econet Filing System

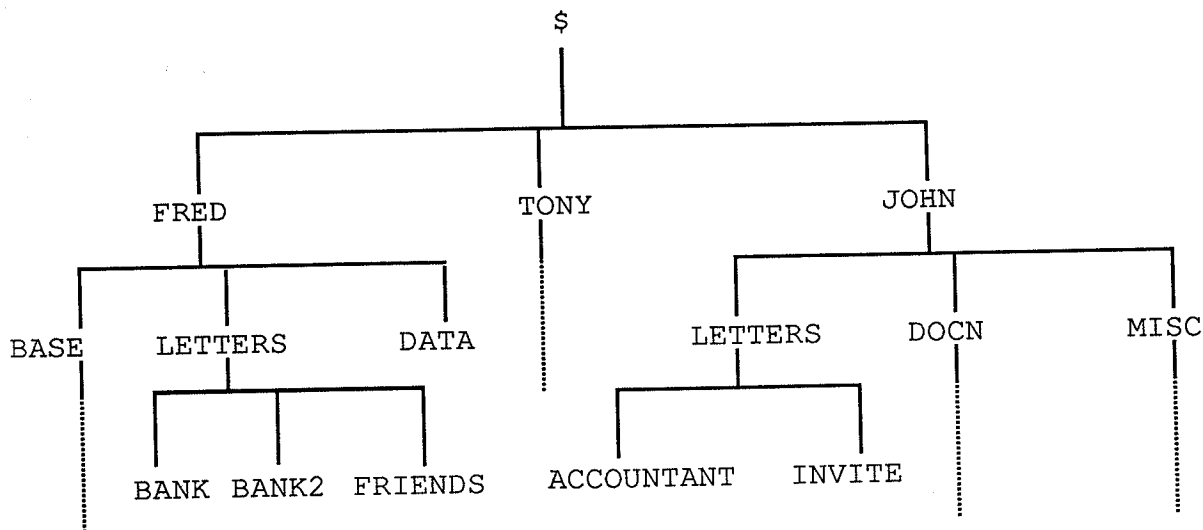
A *file* is the name given to any collection of information that can be stored by the system. The system does not make any distinction between data, program, text or any other files -- it simply stores them as sequences of binary digits, and the significance of them is determined purely by the user and what he does with them. Each file must have a name, so it can be referred to when it is created, read, deleted, etc -- there can be up to 10 alphanumeric characters in an Econet filename (exact specifications are given in Section 3.1.2).

A *directory* is a piece of information kept by the system, so that it can administer files. The directory will contain the names of files in it, and information for the system such as where to find a file on the disc, its length, and so on. Directory names obey the same restrictions as filenames. You can *catalogue* a directory, which means printing out the names of files and other information about them.

### 2.3.1 Hierarchical Directories

In the Econet system (and in many other sophisticated filing systems), directories are themselves files in the system. A directory may contain the names of other directories as well as of files.

This means that a user FRED may have a directory which contains some files (for example BASE and DATA), and also the directories PROGRAMS and LETTERS. The directory LETTERS may contain files GASBOARD and BANK, and perhaps the further directory FRIENDS. Figure 2.1 shows this diagrammatically, as well as some other files.



**Fig. 2.1 A Directory Tree**

This arrangement builds up a *hierarchy* of directories, or if you prefer, a *directory tree*. (Trees in computing almost invariably have their root at the top, and this one is no exception).

Each disc on the File Server has a special *root* directory on it, called \$, which contains all the other directories as *sub-directories*; some of which may be several levels down the hierarchy, as entries in other directories. Each directory or file can be specified by a *pathname* from the root directory; this lists the path down the directory tree taken to reach the entry, with each step separated by the . character. In the above example the pathname for the file BANK would be:

\$ . FRED . LETTERS . BANK

Two items in the network filing system can have the same name, providing they are in different directories (in the diagram there are two directories called LETTERS). However no two items may have the same pathname (attempts to create the second will simply overwrite the first).

Whenever you log on to the File Server you will be put into your own personal directory, called your *User Root Directory* and rapidly abbreviated to URD. This should have been set up for you by the system manager. The next section explains how you can use other directories.

### 2.3.2 Selecting Directories and Using Pathnames

The command **\*DIR** can be used to move between directories in the hierarchy; in general you will need to specify the full pathname of the directory you wish to select next, for example:

**\*DIR \$ . JOHN . DOCN . LETTERS**

After a **\*DIR** command, the directory chosen (if it exists) will become your *Currently Selected Directory* at that station, abbreviated to CSD. It will become the default directory for all directory commands such as **\*CAT**, and the default area used by filing commands such as **SAVE**, when pathnames are not used with the commands. Some commands will only be permitted if you have sufficient access to the appropriate file or directory (see Section 2.4.)

There are several ways of simplifying pathnames: the most important being that, to select a directory below the CSD, you need only specify the pathname starting from the CSD. Thus you can select the sub-directory LETTERS of the directory DOCN, when DOCN is your CSD, with the command:

## **\*DIR LETTERS** (missing out the \$.JOHN.DOCN)

The *wild card* specifiers, \*, # and . can be used in filenames and pathnames in order to save typing, or if you are only sure of part of the name. \* will match with any number of characters in a name, so that, for example, PROG\* could stand for PROG, PROG1, PROGRAM or PROG57B. # will match with any single character, so that FILE# could refer to FILE1 or FILEA; and . can be used at the *end* of a pathname, with the same effect as \*. Wild card specifiers cannot be used when you are giving a name to an entry, as the filename for a SAVE command.

When used in directory names, wild cards that match with more than one possible directory will select only the first matching directory (alphabetically). For filenames, wild cards will specify either the first matching file or all matching files, depending on the operation being carried out on the file.

Note that care should be taken when using \*DELETE with a wild card, as all matching items will be deleted. The System Manager may have set up an option to protect you against deleting too many files, in which case you will have to type \*ENABLE before using a wild card \*DELETE (see Section 3.3).

Full information about wild cards is given in Section 3.1.3. Note that \* commands can be shortened by using the character '.', for example \*DE. will be read as \*DELETE and \*. as \*CAT. Rules for these contractions are given in Section 3.1.1, but they should not be used in programs as additional commands may be added to the system, making the abbreviations ambiguous.

There are other special symbols that can be used in pathnames: & stands for the pathname of your User Root Directory, @ for the Currently Selected Directory, and ^ refers to one level up the hierarchy from the CSD (^.^ gives two levels up). \$ can be used with a disc name to refer to another disc on the File Server, and : is an exact synonym of \$. These symbols are discussed in Section 3.0.2 and disc names are covered in Section 3.0.4.

Commands which are used with a filename, such as SAVE and LOAD, can be used with a pathname in order to specify files which are not in the CSD. For example:

```
LOAD "&.^.^.*"
```

will load the first file from the grandparent of your URD, assuming that you have sufficient access to the relevant file or directory (see Section 2.4).

Commands which operate on a directory, such as \*CAT, can be used with a pathname in order to find out about directories other than the CSD. Using \*DIR without a pathname will return you to your User Root Directory. You may need to use @, when using programs which ask for a directory, to specify your CSD.

The command \*PATHNAME will print the pathname of your CSD, including the disc name of the currently selected disc, for example:

```
*PATHNAME
:MATHS-DISC.$FRED.CHOCOLATE.BISCUIT
```

The command \*CATALL will catalogue a complete tree of directories and files, starting from the CSD. This command can also be used with a pathname.

The command \*CDIR can be used to create a new sub-directory, if you have sufficient access to the parent directory. Unwanted sub-directories can be removed with \*DELETE used with the directory pathname, provided that they contain no entries and are not locked.

A full description of all these commands is given in Section 3.3.

## 2.4.Accounts and Access

All network filing systems need to store information about who should be allowed to read, alter or delete any given file; and how much disc space may be taken up by each user. Econet filing systems allow you to define access rights for each file, which specify what operations on the file are permitted to two levels of users -- owner and public. SJ Research File Servers also support a system of accounts, which clarify the concept of owner access and control the allocation of space on the disc.

### 2.4.1 The Accounting System and Getting Information about Files

There are 1024 accounts on an SJ Research File Server, numbered in hexadecimal from 0 to 3FF. Each file and directory will be associated with a *main account*; and the storage space taken up by the file will be debited from that account. When the item is deleted, or transferred to another account, credit will be restored to the old main account. Each user will be allocated one or more accounts by the System Manager, who will also control the maximum amount of disc space available to each account, by using the commands **\*CREDIT** and **\*DEBIT**.

The command **\*STATEMENT** will give a list of all the accounts to which you have access, and the credit balances for each. The credit balance of an account is in units of 1 kilobyte (K), and represents the amount of space available for files (or directories) with the corresponding main account number. If your File Server has more than one disc, you will have access to the same accounts on each disc, and a separate statement will be given for each disc. For example:

#### **\*STATEMENT**

```
Disc      0
Account   Balance
    23      252k      7D      1296k
    A1    bankrupt      F0      45k

Disc      1
Account   Balance
    23      1296k      7D    bankrupt
    A1    bankrupt      F0      318k
```

This user has access to accounts 23, 7D, A1 and F0, but he will not be able to create any new files or directories in those accounts which are bankrupt: e.g. account A1 on disc 0.

Every file or directory also an *auxiliary account* as well as a main account; and any user with access to either the main or auxiliary account of a item is given **owner access** to that item. Any other user will be given **public access** only. An item does not use any credit in its auxiliary account.

It is necessary to have owner access to a directory in order to create a new item in that directory; and this new item will be given the same main and auxiliary accounts as the directory in which it is created.

The main or auxiliary account number of your URD will have been set to an account that you own. The command **\*INFO** will give complete information about your currently selected directory, including the accounts to which it belongs. (If your station is fitted with the advanced version of the network filing system ROM, you will need to use **\*INFO @** to have the same effect.) For your URD it will give a display of the form:

```
Diana      Entries=3      Default=WR/r
D/          07jan86 today  11:53 23 (00)
```

Reading from left to right, this tells you the name of the directory, the number of entries that it contains, some information about its access rights (see Section 2.4.2), the date that the directory was created, the date and time at which the contents of the directory were last changed, and finally the account numbers associated with the directory. The auxiliary account number is shown in brackets.

**\*INFO** can also be used followed by a pathname to find out about a file, or a directory other than the CSD.

Slightly different information is given for files, with load and execute addresses, and the length of the file in hexadecimal replacing the number of entries and the default access status given for directories (see Section 3.3 for more details). For example:

**\*INFO \$.JOHN.DOCN.LETTERS.MOTHER**

```
Mother      00000000 FFFFFFFF      002723
WR/r        08jan86 10feb86 20:45 04 (FF)
```

The command **\*EX** used with a directory name will give the same information as **\*INFO** for all the entries in the specified directory, preceded by the same header as for the **\*CAT** command.

If you have owner access to a file or directory, you can change the main or auxiliary accounts of that item with the **\*ACCOUNT** command. You can change the auxiliary account number to any value desired (note that you can lose your owner access this way), but you must own any new *main* account. This is because when the main account number is changed, the cost of the item is transferred from the old main account to the new one.

For example, to change the accounts of an item **GEORGE**, in the CSD, to main account 25 and auxiliary account 66, you would type:

**\*ACCOUNT GEORGE 25 (66)**

This would work if you had owner access to **GEORGE** and also owned account 25. Main and auxiliary access can be changed separately by leaving out the irrelevant parameter, for example:

```
*ACCOUNT TEST 23
*ACCOUNT DOCN (47)
```

Note that to change the accounts of the CSD, you must move up a level in the hierarchy with **^**, as otherwise the system will look for an entry in the CSD of the appropriate name and not find it.

## 2.4.2 Controlling Access to Files

A system of access control is needed for every network filing system, so that unauthorised users cannot read, alter or delete confidential or important files. On an Econet File Server, two levels of access are recognised: i.e. **owner** and **public** access, as defined by the account structure. Each entry in the hierarchical directory structure has a set of access letters associated with it, which specify the access rights; the character **'/'** separates owner access from public access. These access letters are printed after the item's name by commands such as **\*CAT**, **\*EX** and **\*INFO**. The system will give the access letters that apply to the current user in capitals, and the others in lower case.

The following letters may appear in either *owner* or *public* access strings, and apply only to users in the appropriate category:

<i>No letter</i>	No access to the file is permitted until access is changed
<b>R</b>	Read access only -- e.g. <b>LOAD</b> but not <b>SAVE</b>
<b>WR</b>	Read and write access -- but you can only use <b>*DELETE</b> on an item that you own
<b>W</b>	Write access only -- useful for append only files

The following letters describe general attributes of the item:

- L Locked item: protected against accidental deletion or renaming by the owners (public access will never allow these operations)
- P Private item: Invisible to all users except owners. If a non-owner attempts to look at the appropriate directory using \*CAT or \*EX, these items will be listed as ...Private, and no operations can be performed on them.
- D Item is a directory. This access letter cannot be changed, and can only be combined with the letters P and L.
- /spl Item has been spooled to the print queue, and is waiting to be printed (see Section 2.5.6). This access code cannot be changed, and can only be combined with the letters P and L. All print queue files have read access to their owners and no access at all to public users. It is not possible to write to a print queue file. Note that the ownership of a file is altered when it is submitted for printing.
- /prt Item has been submitted to the print queue by \*PRINTOUT (see Section 2.5.6), and is waiting to be printed. This code works in a similar manner to /spl.

Users with owner access to an item can use the command \*ACCESS, followed by an item specifier, to change the access status within the permitted range for the item. Any letters after the character / will apply to public access. For example:

**\*ACCESS DATA WR/R**

will change the access status of the file DATA to read and write for the owners, and read access only for non-owners. Wild cards are allowed in the item specifier, and will cause the command to apply to all items for which it is appropriate. So the command:

**\*ACCESS FRE\* L**

will lock all the items in the CSD beginning with the letters FRE, which can be either files or sub-directories. The characters + and - can be used as many times as desired to add and subtract letters from the access string, instead of re-typing it. For example:

**\*ACCESS PROG\* -/W**

will disallow write access to non-owners of all files in the CSD beginning with PROG, without affecting any other access letters. This command will not affect directories, as they cannot have an access of W.

The characters D and -D can be used as special codes to restrict the range of \*ACCESS commands to only directories or files respectively. Thus:

**\*ACCESS \*A01 -D+P**

will make private all files ending with A01, but will not affect any matching directories, as they do not match the specification -D.

When a new file is created, its access status is set to the *default access* of the directory that it is created in. This status is listed as **Default=** when the commands \*EX or \*INFO are used on the directory, and is inherited by any sub-directories. Default access strings can contain the characters W, R, P, L and /. They can be changed with the command \*DEFACCESS; for example:

**\*DEFACCESS PWR/**

will cause all subsequently created files in the CSD to have access **PWR/** and subsequently created sub-directories to have access **PD/**.

**\*DEFACCESS** can be followed by a directory pathname, and the characters + and - can be used in the same way as with **\*ACCESS**.

## 2.5 Stations and Printing through the Network

In a network system, it is unlikely that individual stations will have their own printers. Instead shared printers will be connected to the network, via a *printer server*. This will be a station on the network, usually the SJ Research File Server, or a BBC Microcomputer.

It is useful to know what kinds of printer server are provided on your network before talking about printing, and so the next section discusses the various commands that give information about the stations on the network.

### 2.5.1 Different Types of Station

The command **\*STATIONS** will list all the machines connected to the Econet network which are currently switched on, giving their station number, type and network ROM version number. It does not show the station you are using. For example:

#### **\*STATIONS**

Station	Type	
254	SJ Research File Server	03.42
045	BBC Micro	03.60
002	BBC Micro	03.34

The command **\*FSLIST** will list all the *active* File Servers on the network, with their types and version numbers. If the installation has multiple networks, File Servers on other networks will be displayed, preceded by their network number. For example:

#### **\*FSLIST**

File Servers/Type

254	SJ Research File Server ver M.97/HDFS
064.200	SJ Research File Server ver 0.91/FDFS

The command **\*PSLIST** can be used to find out the station numbers of the printer servers present on your network. Only printer servers that you are allowed to use will be listed (see Section 3.3). Most networks will have one SJ Research File Server, and will use that as the printer server; but some may use a BBC Microcomputer instead or have several printer servers. After each station number, further information about the printer server will be given. So, for example, you might get:

#### **\*PSLIST**

254	Ready
	simple
	fancy
	nobann
250	Offline
064.200	Busy with 064.100

The previous commands will tell you if these printer servers are SJ Research File Servers or BBC Microcomputers. 250 is not shown on **\*FSLIST**, as it is an BBC Microcomputer, and does not respond as a File Server.

You can find out your own station number, and the File Server and printer server currently selected, with the command **\*CV**. For example:

```
*CV
FS number 254
PS number 235
You are 001.003
OSARGS ver 001
```

## 2.5.2 Selecting a Network Printer

### A Step-by-Step Guide

The BBC Microcomputer assumes that printer output is to be sent to a local printer, unless otherwise specified. The command **\*FX5,4** will select the network printer, but uses a default printer server station number of 235, which may not be a printer server on your network.

Note that you will not be allowed to change the printer selection if there are still characters waiting to be printed on the previous printer. If you need to throw away the contents of the printer buffer, type **<Escape><Ctrl-C><Escape>**.

There are two possible ways to select a suitable printer server on the network. On most networks, there will be one SJ Research File Server and it will probably be used as the only printer server. In that case, you should use the command **\*PRINT**, which will both select the network printer and change your printer server number to that of your current File Server. Each File Server can have up to two physical printers connected to it, although they may not both be available for user's output. The system manager will set up a default order in which these printers will be used.

When you are ready to start printing, you should type the character **<Ctrl-B>**, or use the command **VDU 2**. Any characters which appear on the screen will then also be sent to the selected printer server station. The message **Not listening** will be given if this station number does not exist, or is not running a printer server program. If all the printers that you are set up to use on the selected printer server are already busy printing, the message will be **No reply**.

On an HDFS or MDFS, output waiting to be printed will be stored in a special directory until a suitable printer becomes free; so the printer server will always be ready unless this directory is full, but your output may not be printed immediately. Details of this print spooling are given in Section 2.5.4.

Printing is concluded by **<Ctrl-C>**, or the command **VDU 3**, at the end of the printout. Until the user types **<Ctrl-C>**, a non-spooling printer will remain busy with the user's output and no-one else will be able to use the printer. Eventually the printer server's internal timer will finish the user's printout after a sufficient period of inactivity; in this case some of the characters sent to the printer may be lost.

If **<Ctrl B>** is active the user cannot then log off simply by typing **\* BYE**. The printing job will not be closed and noone else will be able to use the printer. The user must type **<Ctrl C> \*BYE** which can be done by **\*LOGOFF** (a non syst.command).

So, to print a listing of the current program on a printer connected to your File Server, the process should go:

```
*PRINT <return>          (to select the printer server)
LIST <Ctrl-B> <return>
....listing of program on screen and printer....
<Ctrl-C>
```

See under **\*TYPE** in Section 3.3 for how to list text files on the screen, and **\*PRINTOUT** in Section 3.3 for a more sophisticated method.

To stop printing temporarily, do *not* use **VDU 3** and **VDU 2**, since this will result on the print job being ended, then another one begun on a fresh page with another header. Refer to Section 2.5.3 for how to use **\*FX3,<number>** to do this.

If you need to use another SJ Research File Server or a BBC Microcomputer as a printer server, you will need a command other than \*PRINT. Instead you should use \*PS, which selects the network printer and broadcasts for a printer server on the network. \*PS can be used with a station number to select that station as the printer server; or on its own, where it will give a display of the form:

```
Printer 200 jammed
Printer 180 ready
Printer 180 selected
```

\*PS will attempt to select a printer answering **ready** for preference, followed by **busy**, and finally **jammed**. If no suitable printer responds, then the message **No station responding** will be given. Printer servers which have no printers you are allowed to use will not respond to \*PS.

If you select a printer server which has a different number from that of the File Server at which you are logged on it will attempt to log you on as ANONPRINT or if this option is not available as the default user. Some SJ Research File Servers, especially those with print spooling, will be set up not to allow printing by users who are not logged on to that File Server. These will not respond to \*PS, and will give the **No reply** message if selected by station number. If you have a User Id on this other File Server, you can log on to it by typing:

**\*I AM <station number> <appropriate User Id and password>**

You can then return to your original File Server by logging on to it again (remember to give the station number), and you will now be logged on to both File Servers, but only for the purposes of printing. Note that you will be returned to your URD on the original File Server with your default options.

Once you have selected a printer server with \*PS, printing proceeds in the same way as for \*PRINT. There are however two possible extra things you may have to be careful about.

The first is that the advanced network ROM (ANFS) contains its own version of \*PS, which will be selected in preference to the network version. This ANFS copy does not perform the \*FX5,4 to select the network printer; so in this case you need to type \*/PS, which will find the network version. The command \*HELP will list all the ROMs fitted to your station, and will say **Advanced NFS** if appropriate.

However the ANFS ROM also contains the command \*FS, which allows you to change the File Server number stored in the BBC Microcomputer. This can be used to change between two File Servers if you are logged on to more than one at once. This command is not available on earlier network ROMs.

The second problem is that some printers do an automatic line feed after every carriage return, and some do not. Your system manager will configure \*PRINT and \*PS to allow for this. However, if you have a mixture of printers on the network, it may still be necessary to type \*FX6,0 before sending output to certain printers, in order to get the right number of line feeds. You should be told if this is required.

## **A more detailed description**

Initially the BBC Microcomputer will assume that the station number of the printer server is 235, and it will attempt to send output to this station if printing is requested. It will thus be necessary to change this printer server number to that of the station you wish to use.

SJ Research File Servers also allow two physical printers, one serial and one parallel, to be connected to the printer server. Each physical printer can have up to four *banners* defined for it; a banner is a text string set up by the system manager to identify each user's output. Thus a typical banner will contain form feeds and such information as user identifier, time and date of printing. Each physical printer/banner combination is set up as a *logical printer*.

Thus two parameters control the network printer selected; the printer server selected and the matching logical printer selection.

The printer server station number is stored in the BBC Microcomputer, and will not be checked until printing is attempted. This number can be changed with the commands \*PRINTER, which sets it to your currently selected File Server number, and \*PS <number>, which sets it to the number specified. The

command **\*CV** will show your current printer server number. This number can include a network number, so that a printer server on another network can be specified.

If printing is attempted to a station that is not running a printer server program, the message **Not listening** will be returned after a pause. This message can also be given if the printer server is unable to print your output for any reason. This case may also give the message **No reply**.

The logical printer selection for each user is stored on each File Server for the appropriate printer server. BBC Microcomputers running printer server programs have only one logical printer, called **PRINT**, so they do not need to store a selection. A default selection is set up by the system manager, and this is given to users when they log on. This default selection is the only one available to users who are not logged on to the File Server.

The commands **\*PRINTER** <logical printer name> and **\*PS** {<logical printer name>} can be used to change your logical printer selections. **\*PRINTER** affects only the selection on your currently selected File Server; when used without a parameter it displays your current logical printer on that station. There are several reasons why you may not be allowed to use and hence to select a particular logical printer; these are discussed later in this Section.

**\*PS** <logical printer name> will broadcast for a printer server on the network which has that particular logical printer. Your internal logical printer selection will be set to this name on all the printer servers which respond, i.e. those which have a logical printer of the appropriate name which you are allowed to use. The printer server number in your station will be changed to match one of the stations which respond. The current status of the printers responding will determine which station is selected; the order of preference is first ready, then busy, and finally jammed. Note that the status of the printers may change before you get around to sending output. If no suitable station responds to **\*PS** <name>, the message **No station responding** is given.

The command **\*PS** with no parameter will work in the same way, except that no logical printer selections will be altered. The command **\*PSLIST** will list the printer servers which will respond to **\*PS**, with their logical printers and the status of your matching logical printer selection.

A particular logical printer may be unavailable for several reasons, including the following ones. A printer may be set to be non-existent; either because it has been disconnected, or because it is reserved for printing special system messages. The system manager may set up an option so that users without access to a particular account cannot use a certain logical printer. The default logical printer on a File Server can be set to only allow users who are logged on, so that anonymous users cannot use that printer server at all.

A logical printer that is not available to you cannot be your selection, unless it is the default, or the printer details have been edited since it was selected. Thus using **\*PS** will find you a suitable printer automatically.

Logical printers on Modular Disc File Servers can be set up to be either *print-spooling* or *non-spooling* by the system manager. Non-spooling printers will only accept output when the matching physical printer is free.

When the physical printer for a print-spooling logical printer is busy, output directed to it is spooled as a *print job* file into the print queue directory on the File Server. Access to a file server account is required for output to be transferred to the print queue, and hence these logical printers are only available to users logged on to the appropriate File Server. Print jobs are printed later when a suitable physical printer becomes free. A full description of the print queue directory is given in Section 2.5.6.

There are two special logical printers, called **HOLD** and **AUTO**. **HOLD** will keep a job in the print queue indefinitely, until the logical printer selected for it is changed with the command **\*REROUTE** <print job name> <logical printer name>. This command can be applied by users to any jobs in the print queue to which they have owner access.

**AUTO** is set up by the system manager to be the most suitable logical printer for users who have no preference. This option may select a sequence of logical printers; so that when printing is attempted, a search down the list is made for a permitted logical printer. **AUTO** and **HOLD** are common choices for the default printer on a File Server.

### 2.5.3 Turning the Printer On and Off

To start printing the user should print the character <Ctrl-B>, either by typing it, or by VDU 2 from a program. The printer will usually be set up by your system manager to print a *banner*, which will be a series of characters designed to be easily visible, followed normally by the user's name, the station number, time and date.

The characters sent to the printer will then be printed. The BBC Microcomputer always filters one character value out, by default this is character &0A (line feed), so that printers that do an automatic line feed after every carriage return can be used. To change the character that is filtered, type **\*FX6,<character>**. If the printer does not have auto line feed, then **\*FX6,0** is a useful choice. The program **\*PS** will perform this **\*FX6,0** automatically if the printer(s) in your network require it (the system manager will configure **\*PS** to match the printer). If your network has two different sorts of printers, with different line feed defaults, then you will need to type **\*FX6,0** yourself for the non-automatic printer.

Printing is concluded by printing <Ctrl-C> (VDU 3) at the end of the printout. The printer will usually be set up to leave a blank page, and will then be ready for another user's output. Until the current user types <Ctrl-C>, the printer server will remain busy with this user's output unless its internal timer finishes the user's printout after a sufficient period of inactivity. Note that this if the printer server times out in this way, some of the characters sent to the printer will be lost.

To stop printing temporarily, do *not* use <Ctrl-B> and <Ctrl-C>, since this will result in the print job being ended, then another one begun on a fresh page with a fresh banner. Use **\*FX3,<number>** to do this, where <number> is a single byte with each bit having the following effect:

Bit 0 (1)	1=enable RS423 (serial) output
Bit 1 (2)	1=disable output to VDU
Bit 2 (4)	1=disable printer output
Bit 3 (8)	1=enable printer (but it must have been started with <Ctrl-B>)
Bit 4 (16)	1=disable output to any *SPOOL file
Bit 5 (32)	no effect
Bit 6 (64)	1=disable printer output, except for characters preceded by <Ctrl-A>
Bit 7 (128)	no effect

For example, calling **\*FX3** with bits 3 and 1 set to 1's (**\*FX3,10**) will disable the VDU and enable the printer, i.e. will print only. Calling with bit 2 = 1 (**\*FX3,4**) will send output to the VDU only (even if <Ctrl-B> has been typed). Calling with no bits set (**\*FX3,0**) will send output to both VDU and printer (this is the default setting).

For printing graphics to the printer only (where it is important that all characters, including the one specified in the **\*FX6** call, are printed), use the following in your program:

```
VDU2      start the printer
*FX3, 0    unnecessary unless you have previously called *FX3
VDU1,<character>
```

VDU1 (<Ctrl-A>) sends the immediately following character to the printer only, but the VDU output must be active for it to work (!) The character following <Ctrl-A> will be sent to the printer, regardless of whether it is the character filtered out (set by the **\*FX6** call; see above)

For programs that do a lot of switching between printer only and VDU only, the following calls are recommended:

<b>VDU2</b>	start the printer
<b>*FX3, 64</b>	this disables printer, except for chars preceded by VDU1
<b>VDU &lt;character&gt; or PRINT &lt;anything&gt;</b>	sends to the VDU only
<b>VDU1,&lt;character&gt;</b>	sends to the printer only

This saves multiple vast numbers of **\*FX3** calls if there is a lot of switching between VDU and printer.

Please note that the documentation of \*FX3 in the Advanced User Guide for the BBC Micro, page 119, is misleading. Line 7: in Econet it is always necessary to send <Ctrl-B> to start printing. Lines 18-20: untrue. You may use Osbyte &EC to read the current state of the bits, but Y will not work as a bit mask for Osbyte 3 !

Please use this \*FX3 call rather than using VDU21 and VDU6 to turn the screen on and off. There are bugs in this part of the VDU driver which will cause undesirable results.

## 2.5.4 Direct Printing of files from the File Server

There is a facility available to print the contents of a file off-line, if desired. The command is \*PRINTOUT, and there is a corresponding command \*PRINTER to select the printer at which the output appears.

There are two advantages to the use of \*PRINTOUT. First, the job can proceed without using any processing power of the BBC Microcomputer. Second, the system is not restricted by the defined printer protocol, and can therefore give much more information back to the user.

See the full description of these commands in the next Section.

## 2.5.5 Banners and Logical Printers

Usually a *banner* will be printed before each user's output: this is a text string set up by the system manager, which may also contain information (such as user identifier, time, date etc.) inserted into the string by the system.

An example of a banner is:

```
SJ Research File Server *** Station 5 (FRED) 08feb86 13:23:04 ***
```

The banner file also contains a standard string to be added at the end of each user's printout. An example could be a row of asterisks followed by a page throw.

Each physical printer on an SJ Research File Server may have up to four different banners available, and these are distinguished as different *logical printers*. Thus there may be up to eight logical printers on each SJ printer server, and their names are listed after the station number by \*PSLIST.

Printer servers which are BBC Microcomputers will have only one logical printer, called **PRINT**, and this name will not be listed by \*PSLIST. (\*PS and \*PS PRINT have an identical effect.)

There is a default for the logical printer selection on each File Server, set up by the system manager and selected for you from when you log on to a station until a particular printer is specified. The system manager will also set up an automatic printer selection, to be used by users who do not have a printer preference. This will select a printer and banner to be used, and may choose the other physical printer if the first choice is busy. The automatic and default printers will usually be the same, or the default printer may be set up to do nothing.

On print-spooling printer servers there are also two special logical printers called **HOLD** and **AUTO**. **AUTO** represents the automatic logical printer in the print queue, and **HOLD** keeps a file in the print queue indefinitely.

The command \*PRINTER will tell you which logical printer is selected for your station on the currently selected File Server, and its current status. For example:

```
*PRINTER
AUTO    : with printer spooling
```

\*PRINTER can be used with a logical printer name to change the logical printer selection. This will change the banner printed with your text, and perhaps the physical printer it comes out on. Some logical printers

may not be available to general users, and so you will not be able to select them.

**\*PS** can also be used with a logical printer name. This will broadcast for a printer of that name and will work in the same way as **\*PS** on its own. Logical printers with the same name on different File Servers should be identical, and preferably near each other.

Note that **\*PS <printer name>** will change the printer selection on all File Servers that respond to it, as it is a broadcast message.

### 2.5.6 The Print Queue Directory

If your File Server is an HDFS or MDFS, it will be able to carry out *print spooling*. This means that output sent to particular logical printers will be stored in a special directory on the File Server if it cannot be printed immediately. It will then be printed when a suitable physical printer becomes free.

Each logical printer will be set to be either *spooling* or *non-spooling* by the system manager. Non-spooling logical printers will only accept output when the matching physical printer is free; this is the only kind of logical printer possible on the FDFS and RM380Z File Servers. Non-spooling printers are useful when the network is quiet if you want to print program output as it is produced.

The directory in which output is stored is called **%PRINTQ**, and is a sub-directory of the lowest numbered disc on the File Server. It is not necessary to specify the pathname or disc of this directory when referring to it. If this directory is not found, or is full, all logical printers will be treated as non-spooling.

When a print-spooling logical printer is busy, output sent to it is spooled as a *print job* file into the directory **%PRINTQ**; this file is labelled with system information and given a new name. These names are given sequentially, starting as AA00, AA01, AA02 etc. and going up to ZZ99. Thus the command **\*CAT**, which lists files alphabetically, will show the order in which entries were submitted. When a physical printer becomes free, the file next printed is the first entry in the catalogue that is suitable for that printer. Thus print jobs are carried out in a sensible order.

The main account number of print job files will be set to that of the print queue directory, so as not to take up user's account credit. This main account will normally only be available to the system manager. When the job is submitted, the system will work out the user's *personal account*, i.e. the highest numbered account to which the user has access. The auxiliary account of the print job will be set to this personal account, so users will have owner access to their own print jobs. Account numbers can be changed in the usual way.

Print jobs are given a special access code of **/spl** to mark that they are waiting to be printed. This access cannot be changed, although the file can be locked or made private. Print jobs which are locked will not be printed until they are unlocked. Writing to print job files is not allowed, but they can be read and deleted by users with owner access. Files created with **\*PRINTOUT** (see Section 2.5.4) are given access code **/prt**, and the same conditions apply to them.

The commands **\*EX** and **\*INFO** will give information about files in the print queue directory, and this will be of a different form to the information given for ordinary directories. For example:

```
*INFO %PRINTQ.AA23
```

```
AA23   DIANA   at Stn. 253 0003A6  
L/spl  HOLD   today 12.02 01 (FF)
```

Reading from left to right, this tells you the name of the print job, the name and station number of the user who submitted it, the hexadecimal length of the file in bytes, the access code, the logical printer selected for the job, the date and time of submission, and finally the accounts associated with the file. If you cannot identify particular jobs from this information, you can use **\*TYPE** to show them on the screen.

It is possible to change the logical printer selected for a print job to which you have owner access, using the command **\*REROUTE <print job name> <logical printer name>**. For example:

```
*REROUTE AA23 NOBANN
```

would change the selected logical printer from HOLD to NOBANN in the above example. This will only work if you are allowed to use the new logical printer. The new printer could be non-spooling, in which case it would be treated as spooling with respect to this particular job.

The system manager, and other users with owner access to %PRINTQ, can use the command **\*RENAME** to change the name of a print job, and thus its position in the queue. ! is considered to be the first legal file name character alphabetically, and so is commonly used to start such priority print job names. When the File Server is turned off, the naming sequence is restarted at AA00, and so entries in the stored print queue may be renamed by the system manager.

## 2.6 Other Facilities Available on the Network

This chapter concludes with an introduction to some of the other features provided by the Econet network and the SJ Research File Server. A more detailed coverage is given in Chapter 3, and a complete list of error messages in Appendix A.

### 2.6.1 Passwords, Libraries and Boot Files

You will probably want to change your password from that given to you by the system manager, both to make it more memorable and to protect your files from unauthorised users. The command **\*PASS <old password> <new password>** allows you to do this. For example:

**\*PASS qwerty breakfast**

will change your password from **qwerty** to **breakfast**. If no password has been set, it is necessary to quote a null string "" as the old password. Passwords may contain up to 10 characters -- permitted characters are letters, numbers and ! - \_ . If security is important to you, do not use a password that is easy to guess, e.g. your telephone number or your boy/girlfriend's name.

The system manager may set an option to prevent you changing your password, for example if your User Id is shared between several people. If you forget your password, the system manager will be able to find it out for you.

Another directory selection is stored for each station by the File Server, as well as the URD and CSD. This is the *currently selected library*, which is an option set up for you by the system manager and shown in the third line of the header to the **\*CAT** command. If a file was not found for a command like **LOAD "<pathname from the CSD>"**, the same search for the file will be made from the library directory. Files found there will be acted on as usual.

A complete list of the commands for which library searches are made is given in Section 3.3 under **\*LIB**. Note that **\*<file specifier>**, which loads and runs a machine code program, will search the library, and this is how many commands used on the File Server are stored. Those commands listed as programs in Section 3.2 are kept in the utilities library. However you may want to change your library directory at a particular station, perhaps to use a library with extended utilities provided for a Master Series machine, and you can do so using the command **\*LIB**. For example:

**\*LIB \$.NEWLIB**

will select the directory NEWLIB as your current library directory. When you log on again, your library directory will be returned to the default selection.

It is also possible to set up a sequence of commands to be executed automatically when you log on. The command **OPT4,<number>** controls this, with the number between 0 and 3. Your current selection is shown in the second line of the header for the **\*CAT** command, and is referred to as your boot option.

**\*OPT4,0** will give no action at log on, but the other options will search for a file called **!BOOT** in first your URD and then your library directory. **OPT4,1** will load this file into memory, **OPT4,2** will run it as a machine code program, and **OPT4,3** will read the file as though it were typed in at the keyboard. The last of

these is a useful option as it allows all types of commands to be included in the boot file.

If the only file !BOOT in the system is in the library, then a message of the day can be produced at log-on for every user with OPT4,3 set. The system manager may lock your boot option so that you cannot change it.

You can create a !BOOT file for use with \*OPT4,3 for yourself, by using the command **\*BUILD <file specifier>**, which creates a file of the appropriate name and then prompts for keyboard input, which is sent directly to the file. Pressing the <Escape> key will end the file. For example:

```
*BUILD !BOOT
```

```
0001 *| Hello Diana
0002 *CAT
0003 *CV
0004 <Escape>
```

will create a simple boot file. **\*|** is the operation system equivalent of the BASIC statement REM, i.e. it causes the rest of the line to be ignored. Files built in this way can be edited using a suitable text editor such as WORDWISE.

The command used by \*OPT 4,3 is **\*EXEC <file specifier>**, which reads a text file as if it were typed in at the keyboard. This is useful for performing sequences of commands repeatedly, or for converting a text file into a BASIC program. BASIC programs are usually stored in a condensed form by the language system. Useful subroutines and procedures for facilities like graphics can be provided in \*EXEC format, and you then use this command to add them onto your BASIC programs.

To convert a BASIC program to text, you can use the command **\*SPOOL <file specifier>**, which sends all text from the screen to the specified file. Typing **\*SPOOL** on its own will close the file. For example:

```
*SPOOL PROGLIST
```

```
LIST
```

```
      listing follows here
```

```
.
```

```
.
```

```
*SPOOL
```

will send a listing of the current program, preceded by a line saying LIST and ended by a line saying \*SPOOL, to a text file called PROGLIST. \*SPOOL may be useful with the command \*PRINTOUT, and this is discussed in Section Whatsit.

Note that these commands send information over the network in single byte packets, which is inefficient. Running the utility program **\*PUTGET** will collect these packets together into blocks of 64 bytes, which results in a considerable speed increase. The use of \*PUTGET is thus normally recommended.

A more detailed description of all these commands is given in Section 3.3.

## 2.6.2 General Information Available on the Network

There are several other commands which provide information to users of the network, as well as those discussed earlier in this chapter. Again a complete list is given in Section 3.3, but the simpler commands are covered here.

The command **\*TIME** prints out the time and date on the screen, from the *real time clock* contained in the File Server. There are also versions suitable for incorporating into users' programs, called **\*PTIME**, **\*PDATE**, **\*PDATE2** and **\*GTIME**.

The command **\*VERS** displays the version number of your currently selected File Server.

The command **\*FREE** gives a list of all the discs present on the File Server, and the amount of storage space left on each disc.

The command **\*USERS** lists all the users currently logged-on to the File Server, their station numbers and whether they have system privilege i.e. can carry out operations only normally available to the system manager. A user may appear on the list several times, if he is logged on with more than one station. The list is re-ordered every time a filing system operation occurs, so that the station that performed the operation is moved to the top of the list.

There are special users called **\*-SPOOL-\*** and **\*-SYSTEM-\*** which are given by **\*USERS**: they are used by the system to carry out print spooling and other system operations.

The commands **\*PUSER** and **\*GUSER** can be used to incorporate the current User Id into users' programs. These are discussed in Section 3.3.

### 2.6.3. Copying Files and Directory Structures

If a file is to be *moved* between directories within one disc on a File Server, it is most convenient to use the command **\*RENAME** with appropriate pathnames. However to copy a file, or move between discs or filing systems, the BASIC program **COPIER** is necessary. The program is started by typing:

```
CHAIN "COPIER"
```

The program will prompt for a source and destination filing system. To copy between two different filing systems, enter the names of the systems, e.g. **\*DISC** and **\*NET**. To copy between two File Servers, type **\*IAM** **<File Server station number>** **<User Id>** [**<password>**] **<Return>**; and to copy between two directories, enter two directory pathnames.

You will then be prompted for the name of the file to be copied, and the new name required on the destination filing system. If directory pathnames have been specified they need not be given again. Typing **<Return>** when asked for the new file name will assume the name is unchanged. Note that only one level of sub-directories below the root is permitted by the disc filing system, and that these sub-directories must have single character names. Attempting to copy more complicated paths may cause problems with illegal file names at a later date.

The program will continue to prompt for files to be copied until **<Escape>** is pressed. The user must obviously have sufficient access to the files and directories involved.

There is also a BASIC program called **MULTICOPY** which can be used to copy entire directory trees between File Servers. This works in a similar way to **COPIER** and is explained in Section 3.3. A BASIC program called **ERAQ** is also described, which will delete all or part of entire directory trees. Note that this will delete files even if they are locked.

### 2.6.4 Communicating with Other Network Users

It is possible to use the Econet network to send messages to other users. One of the simplest ways to do this is with the command **\*NOTIFY**. This can be used with a station number, for example:

```
*NOTIFY 4 Merry Christmas
```

This will cause the message **\*| 023: Merry Christmas** to be printed on the screen, where 23 is the number of the station sending the message. The message will be accompanied by a beep.

**\*NOTIFY** can also be used with a User Id, in which case it will be sent to the station at which that user last performed a filing system operation. This will be the station listed first for that user by the command **\*USERS**.

Attempting to send a message to a user who is not logged on, will give the error message **Not logged on**. Sending to a station which is not connected to the network will give the message **Not listening**. It is not possible to send a message to yourself.

If you do not wish messages from other users, you can use the command **\*PROT** to prevent your station

from responding. **\*UNPROT** will remove this protection.

The command **\*VIEW** allows a user to make a complete copy of a remote station's screen. If the remote station is in a screen mode that uses more memory than your current selection, the error **Mode x** will be given, where x is the screen mode of the remote machine. The command prompt will be returned and commands can be entered as usual.

The command **\*REMOTE** allows a user to take over a remote station, so the screen of the remote station echoes the screen of the controlling station. This is useful for demonstration purposes, but will interrupt any work in progress by the user of the remote station and so should be used with care. Ordinary users may not be given access to **\*REMOTE**. The command **\*ROFF** will turn off the remote control.

Both **\*REMOTE** and **\*VIEW** can be used with a station number or a User Id to specify the remote station. The same rules as for **\*NOTIFY** are used to find the station appropriate to a User Id. **\*PROT** can be used to prevent either of these commands from affecting your station; this is obviously essential if you are doing anything confidential. All these commands are discussed in greater detail in Section 3.3.

# Chapter 3:

## Utility Programs Reference

---

This Section gives a full description of the high level interface to the Econet system in BBC Microcomputer using the SJ Research (or Acorn) Econet system.

The chapter is split into several sections, as follows:

- 3.1 Introduction
- 3.2 Summary of Commands
- 3.3 Command Details

### 3.1 Introduction

Some of the commands described in section 3.3 are interpreted directly by the BBC Microcomputer operating system, the local NFS software, the BASIC language or the File Server, others are *transient programs*, which are executed in a designated area of system workspace, and will not corrupt any main program: for example the program **\*TIME** or its variants will not affect the normal operation of a BASIC program.

Other programs in this Section are BASIC programs, and will overwrite any other program present -- but these are mostly utility programs which do not need to run with another program: for example **COPIER** which copies files from disc to network etc. They can of course be merged with other programs if desired.

The title *Command Type* and the suggested syntax will tell you which type of program or command each one is. The transient and utility programs are supplied in the system library (directory **\$.LIBRARY** as supplied). To use one of the utility programs in the library, type just the command, for example:

**\*TIME**

There must *not* be a program with the same name in the currently selected directory. If there is such a file, it will be necessary to type:

**\*\$.LIBRARY.TIME**

The mechanism of library searching is described under the **\*RUN** or **\*LOAD** commands.

In the case of transient or utility programs, there may be some described which are not available on your network system. This is because they would not be required for the type of work being done in your establishment, and so your System Manager will not have supplied them in the library.

### 3.1.1 Syntax Definitions

For each command, the syntax is given in Backus-Naur form with the extension that braces {} indicate optional repetition (or omission) of an item. Backus-Naur form is introduced in the BBC Microcomputer User Guide, Section 33.

The following notation is used:

: := means "is defined as"  
| separates between alternatives  
[ ] means that the enclosed item is optional  
{ } means that the item type may be repeated or omitted  
< > means that the enclosed item is a term defined elsewhere

Note that all "\*" commands may be abbreviated in the BBC Microcomputer system, for example \*DE. will be read as \*DELETE.

If there is any ambiguity arising from the use of a command or an abbreviation, these rules are followed:

The BBC Microcomputer Operating System will check its own table of commands, and then those of any language or filing system ROMs in the computer. Hence \*D. will be read as \*DISC (assuming disc system fitted), and not as \*DELETE.

The command will then be passed to the File Server to check its table of commands. If the command is not a File Server command, then the File Server will search first the currently selected directory (CSD), then the library, for a file of the same name as the command. The BBC Microcomputer will then \*RUN this program.

If the command typed is \*<characters>. (note the dot after <characters>) the File Server will search the CSD, then the library, for the first match to <characters>\* and \*RUN this program. See section 3.0 on wild card specifiers for full details.

If you have a program which has the same name as one of the system commands (or you are not sure), use the \*RUN command or its abbreviation \*/ to ensure that the program is run. An example of this is the program \*TYPE - computers which have a DFS fitted will execute \*TYPE from ROM. In the File Server library, there is a version of \*TYPE which runs much faster on the network. To be sure of using the right version, use

\*/TYPE <file specifier>

\*\<command> Forces the command through to the file server as one of its internal commands eg. DELETE, RENAME and PRINTER etc. Note that \*CAT and \*EX are not commands internal to the file server.

Note that the command \*| (vertical bar) introduces a comment. This is useful in EXEC files, to introduce comments or messages.

Notes : \*I. will be read as \*INFO, but \*I . -- note the space between I and the dot -- will be read by the local NFS as \*I AM.

Note also that \*EX is a command in its own right, but that \*EX. will be read as \*EXEC by the operating system.

### 3.1.2 File, Directory and General Specifiers

A file specifier is defined as:

<file specifier> ::= [<directory specifier>.]<name>

where

<directory base> ::= :[<disc name>] || \$[<disc name>] || & || @ || ^

<directory specifier> ::= [<directory base>.] {<name>.<name>}

A general specifier is either a file specifier or a directory specifier, and is defined as:

<general specifier> ::= <file specifier> || <directory specifier>

Each <name> is of maximum length 10 characters; those allowed are alphanumerics, the characters ! - \_ (underscore), or the wild card characters \* or # (explained below).

Upper and lower case characters are treated as equivalent, so that **File**, **FILE** and **file** all refer to the same file. Whilst some File Servers will accept other characters than those listed above, *it is recommended that programmers use only those in the list, as this will assure compatibility between different versions of File Servers and from different manufacturers.*

The last name specifies the file, and the previous one(s) directories. The directory specifier(s) and dot(s) may be omitted if the file is in the currently selected directory (see under \*DIR for details) or in the current library (see under \*LIB).

The <disc name> and the colon (or \$) may be replaced by \$ or : on its own if the file is on the currently selected disc.

Each user has a *user root directory* (URD), usually having the same name as the user identifier used with the \*I AM command. This directory may contain files and other directories, and these latter directories may themselves contain further directories as well as files. For example:

```
*I AM FRED
LOAD "SYSTEM.DEMO.PROG1"
```

loads a file called PROG1 in the directory DEMO. Directory DEMO is itself in SYSTEM, and SYSTEM is in the directory FRED.

There is a root directory on each disc, which is the directory containing all the user root directories. The name of this directory is the same as the name of the disc, so if the disc is called MAIN1, the full name of the file above is:

```
:MAIN1.FRED.SYSTEM.DEMO.PROG1
```

This directory hierarchy is useful for keeping associated programs or text together. A number of commands can operate on complete directories, allowing time to be saved.

The following abbreviations are available:

- \$ refers to the system root directory on the currently selected disc. In the above example \$ is equivalent to \$MAIN1 (or :MAIN1).
- :
- is an exact synonym of \$.
- ^ refers to one level up the hierarchy. If user FRED had selected **SYSTEM.DEMO** as his current directory with the \*DIR command, then ^A1 is equivalent to **SYSTEM.A1**.
- & refers to the user root directory. In the above example &.FILE2 would be equivalent to \$.FRED.FILE2.
- @ refers to the currently selected directory (CSD). @.PROG1 refers to the same file as **PROG1** on its own, *except that if PROG1 is not found in the CSD, the library will not be searched.* In fact, this form would only be used if the programmer did not want the system to search the library for a file.

### 3.1.3 Wild Card Specifiers

A "wild card" in a file specifier allows reference to a group of files. The following three wild card characters are available:

- # matches any *single* character. Hence PROG# will refer to PROG1, prog2, and ProgC, but not PROG.
- \* matches any number of characters, including zero characters. If there were a directory PROGS in the current directory, and it had three files "XYZ", "A1" and "A2" in it, then \*DELETE PROGS.\* will delete them all.
- . (dot) as the *last* character of the file specifier has exactly the same effect as \* at the end, so that \*DELETE P. will delete all files in the CSD that begin with the letter P.

The following rules apply to wild card characters:

For any operation, if a wild card is used in the directory name then the first directory (alphabetically) will be referred to *only*. For example, if directories A, B and C were in the current directory, then \*DELETE \*.\* will be equivalent to \*DELETE A.\*.

The last part (i.e. file name) in a file specifier may *not* contain a wild-card character in a SAVE, OPENOUT, \*SAVE or \*CDIR command.

In \*DELETE, \*ACCESS, \*RENAME and \*ACCOUNT the use of a wild card in the file name will cause the operation to be carried out on *all* files or directories that match. The system manager can set an option for each user, which requires the user to type \*ENABLE before a wild card delete operation. If this option is set, then the error Not ENABLEed will be given.

In \*DIR, LOAD, \*LOAD, \*RUN and other commands the use of a wild card in the file name will refer to the *first* (alphabetically) of the possible files.

In a multiple match operation, if a filing error (e.g. insufficient access) would occur as a result of the requested operation, then that file (or directory) will be *passed over*, and the operation will continue on the rest. If it was not possible to do the operation on any file, then the error message **Nothing happened** will be displayed.

### 3.1.4 Disc Names

Each disc is given a name by the person in charge of the system when the disc is initialised. A disc name may be up to 10 characters, taken from the same characters as legal file names.

When referring to a file specifier in its full form, note that the colon : or dollar \$ must precede the disc name. This tells the system that the first part of the specifier is a disc name, and not a directory name.

Note that, for example:

\$BOOT or :FRED

are disc names, whereas:

\$.BOOT or :.FRED

''''  
are names of directories in the root of the currently selected disc.

## 3.2 Summary of Commands

This section gives a full description of the commands and utility programs for the Econet system on a BBC Microcomputer. The descriptions are arranged alphabetically. The column **Type** describes which machine holds the program that processes the command, in the case of **File Server command** the File Server executes the command internally and no program is loaded in the BBC computer. If the type is **Transient program** then the program will be loaded into the BBC Computer's RAM at page 9 or page E (see BBC Microcomputer User Guide section 40) and will probably not interfere with any existing program in memory. The types **N.F.S. command** and **A.N.F.S. command** are commands that are processed in the BBC microcomputer but do use workspace in page 9 and page E.

The commands described are as follows:

Command	Type	Syntax
*ACCESS	File Server command	*ACCESS <general specifier> <access string>
*ACCOUNT	File Server command	*ACCOUNT <file specifier>[<main account no>] [<(aux account no.)>]
BPUT#	BASIC keyword	See BBC Microcomputer User Guide (Section 33)
BGET#	BASIC keyword	See BBC Microcomputer User Guide (Section 33)
PTR#	BASIC keyword	See BBC Microcomputer User Guide (Section 33)
EXT#	BASIC keyword	See BBC Microcomputer User Guide (Section 33)
EOF#	BASIC keyword	See BBC Microcomputer User Guide (Section 33)
INPUT#	BASIC keyword	See BBC Microcomputer User Guide (Section 33)
PRINT#	BASIC keyword	See BBC Microcomputer User Guide (Section 33)
CLOSE#	BASIC keyword	See BBC Microcomputer User Guide (Section 33)
*BUILD	Transient program	*BUILD <file specifier>
*BYE	N.F.S. command	*BYE
*CAT	BBC Micro O.S. command	*CAT [<directory specifier>]   *. [<directory specifier>]
*CATALL	Transient program	*CATALL [<directory specifier>]
*CDIR	File Server command	*CDIR [<directory specifier>]
*CLOSE	Transient program	*CLOSE
COPIER	BASIC program	CHAIN "COPIER"
*CV	Transient program	*CV
*DEFACCESS	File Server command	*DEFACCESS [<directory specifier>] <access string>
*DELETE	File Server command	*DELETE <general specifier>
*DIR	File Server command	*DISCS
*DISABLE	File Server command	*DISABLE   *DISABLE SAVES   *DISABLE LIBRARY
*DISCS	Transient program	*DISCS
*DUMP	Transient program	*DUMP <file specifier> [<offset>]
*ENABLE	File Server command	*ENABLE   *ENABLE SAVES   *ENABLE LIBRARY
ERAQ	BASIC program	CHAIN "ERAQ"
*EX	N.F.S. command	*EX [<directory specifier>]
*EXEC	BBC Micro O.S. command	*EXEC <file specifier>
*FREE	Transient program	*FLUSH
*FLUSH	File Server command	*FREE
*FS	A.N.F.S. command	*FS [<network number>.] <station number>
*FSLIST	Transient program	*FSLIST
*GNET	Transient program	*GNET
*GO	Transient program	*GO <32 bit address>
*GTIME	Transient program	*GTIME
*GUSER	Transient program	*GUSER
*HELP	BBC Micro O.S. command	*HELP
*I AM	N.F.S. command	*I AM [<network number>.] [<File Server number>] <User ID> [<password>]

*INFO	File Server command	*INFO [<general specifier>]
*LIB	File Server command	*LIB <directory specifier>
LOAD	BASIC command	LOAD "<file specifier>"   LOAD <string variable>
*LOAD	BBC Micro O.S. command	*LOAD <file specifier> [<load address>]
*LOGON	Transient program	*LOGON
MULTICOPY	BASIC program	CHAIN "MULTICOPY"
*NOTIFY	Transient program	*NOTIFY <station number>   *NOTIFY <User Id> <text> <numeric variable>=OPENIN "<file specifier>"   <numeric variable>=OPENIN <string variable> <numeric variable>=OPENUP "<file specifier>"   <numeric variable>=OPENUP <string variable> <numeric variable>=OPENOUT "<file specifier>"   <numeric variable>=OPENOUT <string variable>
OPENIN	BASIC keyword	
OPENUP	BASIC 2 keyword	
OPENOUT	BASIC keyword	
*OPT1	BBC Micro O.S. command	*OPT1,<number>
*OPT4	BBC Micro O.S. command	*OPT4,<number>
OSCLI	BASIC 2 keyword	OSCLI "<string>"   OSCLI <string variable>
*PASS	File Server command	*PASS <old password> <new passwrod>
*PATHNAME	Transient program	*PATHNAME
*PRINT	Transient program	*PRINT
*PRINTER	File Server command	*PRINTER   *PRINTER <logical printer name> *PRINTER <file specifier>
*PRINTOUT	File Server command	
*PROT	Transient program	*PROT
*PROTEX	Transient program	*PROTEX
*PS	Transient program	*PS <station number>   *PS   *PS <logical printer name>
*PSLIST	Transient program	*PSLIST
*PTIME	Transient program	*PTIME
*PDATE	Transient program	*PDATE
*PDATE2	Transient program	*PDATE2
*PUSER	Transient program	*PUSER
*PUTGET	Machine code program	*PUTGET
*PUTGET2	Machine code program	*PUTGET2
*REMOTE	Transient program	*REMOTE <station number>   *REMOTE <User Id.>
*RENAME	File Server command	*RENAME <old general specifier> <new general specifier>
*REROUTE	File Server command	*REROUTE <print job name> <logical printer name>
*ROFF	N.F.S. command	*ROFF
*RUN	BBC Micro O.S. command	*RUN <file specifier>   */<file specifier>   *<file specifier>
SAVE	BASIC command	SAVE "<file specifier>"   SAVE <string variable>
*SAVE	BBC Micro O.S. command	See command details
*SDISC	File Server command	*SDISC <disc name>
*SPOOL	BBC Micro O.S. command	*SPOOL <file specifier>   *SPOOL
*STATEMENT	Transient program	*STATEMENT
*STATIONS	Transient program	*STATIONS [<network number>]
*TIME	Transient program	*TIME
*TYPE	Transient program	*TYPE <file specifier>
*UNPROT	Transient program	*UNPROT
*USERS	Transient program	*USERS
*VERS	Transient program	*VERS
*VIEW	Transient program	*VIEW <station number>   <user identifier>

**Syntax:** \*ACCESS ||B<general specifier> <access string>

## Action with Wild Cards in the File Name:

Occurs on every match. Note the special use of access letter D below.

## Description:

This command allows the access status of a file to be changed. New files are created with default access status **WR/R**, unless this is changed using the \*DEFACCESS command (see this Section).

An *owner* is defined as someone with access to the account or the auxiliary account of a file (or directory). A *non-owner* has *public access* to a file (or directory).

The access string is of the form:

[<owner access>][/<public access>]

where <owner access> and <public access> are strings of letters taken from the list below. If no characters appear in either string, then this signifies no access to the file (or directory) for that category of user. If no '/' appears, then all letters will be taken as defining *owner access*, and *public access* will be none.

Access letters which apply to the current user will be given in capitals, and the others in lower case.

## Letters that can appear in either *owner* or *public* strings

**No letter:** No access to file until status has been changed.

**R** Read only access: the file can be saved over, loaded, or opened for reading using OPENIN (see this Section).

**WR** Read and write access allowed: this means that the file can be loaded, or opened for update using OPENUP (see this Section)

**W** Write only access: this means 'append to end only'. The file can be opened using OPENUP, and written to, but only if PTR# = EXT#. An example of its use is keeping a record of users of a program, without the users being able to read it.

## Letters that describe general attributes of the item

**L** Locked item: cannot be deleted, saved over or renamed until access is changed. Users with *public access* can never delete or rename a file, so this applies only to *owners*. A directory cannot be deleted until it contains no entries, so locking a directory is likely to be useful only to prevent renaming.

**P** Private item: invisible to anyone but the *owner*. If a *non-owner* attempts to look at the appropriate directory using \*CAT or \*EX, these items will be listed as ...**Private**, and if he attempts to perform any operation on the item, the error message **Not found** will be given.

**D** Item is a directory. An attempt to change this access letter will cause an Error 46 (see below), but it may be used to specify directories in wild card operations (see below).

**/spl** Item is waiting to be printed. This access code is given automatically to entries spooled to the print queue directory, %PRINTQ, and cannot be changed. *Owners* have read access to these entries and *non-owners* have none.

**/prt** Item is waiting to be printed. This access code is given automatically to entries generated in %PRINTQ by the \*PRINTOUT command (see this Section), and cannot be changed. These

entries contained pointers to the file to be printed out; *owners* have read access only to this information if they wish.

### Additional characters that may appear in the access string

- +** adds the following letters to the existing access status
- subtracts the following letters from the existing access status.
- /** separates *owner access* status (before the /) from *public access* status (after the /). There may not be more than one '/' in the access string.

The letters **L P** and **D** may be specified before or after the '/', but will appear before it when the file is listed in a \*CAT or \*EX command. There may be any number of '+' or '-' signs in the access string.

Wild cards are permitted in the file specifier, and will cause the command to apply to all matching files (or directories). The usual rule concerning wild cards applies; the operation will be applied only to files (or directories) where an error would not be caused. For example:

```
*ACCESS Data* D+P
```

will make private all directories beginning with the letters DATA, but not change any files (since the attempt to give the access letter **D** to a file would cause an error). There is however in this command an important extension to this rule, namely that **-D** can be used to specify a file not a directory. The command

```
*ACCESS Data* -D+/W
```

will give write access to *non-owners*, of all files (only) beginning with the letters DATA.

Note that, to change the access letters of the user's own user root directory, it will be necessary to type (for example):

```
*ACCESS ^.FRED +P
```

### Random Access Operations

The effect of the access status on random access operations is shown in the tables below:

Access Status	Operation		
	OPENIN	OPENUP	OPENOUT
D	D	error INAF	error INAF
none	error IA	error IA	WR
W	error IA	W	WR
R	R	error IA	WR
WR	R	WR	WR

Table 1: Effective access after different file opening commands.

Effective Access	Operation		
	BGET#	BPUT# Read EOF	Set/Read PTR#
D	error INA	error INAF	error INAF
W	error IA unless PTR# = EXT#	error IA	OK
R	OK	error NOFU	OK
WR	OK	OK	OK

Table 2: Operations allowed for each type of effective access.

Error IA is the error message "Insufficient access" (Error BD)

Error INAF is the error message "xxx is not a file" (Error B5)

Error NOFU is the error message "File not open for update" (Error C1)

## Examples:

\*ACCESS Prog22 WR/

will change the access status of file Prog22 to read and write for the *owner*, but no access at all for *non-owners*.

\*ACCESS Prog\* +L

will find every file (or directory) beginning with the letters PROG, and will lock each one.

\*ACCESS \* D+P

will find every *directory* within the currently selected directory and add the letter P to their access status, i.e. making them all private access.

\*ACCESS \* -D+P

will find every *file* within the currently selected directory and add the letter P to their access status, i.e. making them all private access.

## Likely Errors:

**FS Unusual Error 46**                      **Error 168 (&A8)**

An attempt to set an illegal attribute (e.g. W to a directory or D to a file).

**Insufficient Access**                      **Error 189 (&BD)**

Only an *owner* can change the access status of a file.

**Bad attribute**                              **Error 207 (&CF)**

Attempt to use letters other than P D W L R.

## Associated Keywords:

\*DEFACCESS

## Compatibility Notes:

This command is supported by Acorn systems, but access letters P and /spl are not, nor are the '+' and '-' characters. Note also that Acorn systems define file ownership differently, and that there is no \*DEFACCESS command.

---

**Syntax:** \*ACCOUNT<file specifier> [<main account no.>] [(*<aux account no.>*)]

## Action with Wild Cards in the File Name:

Occurs on every match.

## Description:

This command changes the main account number of the file specified to <account no.> and/or the auxiliary account number to <aux account no.>. Account numbers are three digit hexadecimal numbers, ranging between 0 and 3FF.

When a file or directory is initially created, it is given the same main account number and auxiliary account number as that of the directory that it is in, and the space taken by the item is debited from the balance in the main account. As explained in Section 2.4, a user has owner access to an item if he has access to either the main or the auxiliary account, otherwise he has only public access.

When the *main* account number is changed, this command debits the space taken by the item from the 'new' account, and credits the same amount to the 'old' account.

The user must be an owner of the item, i.e. he must have access to either the main account or the auxiliary account of the file (or directory). If he is changing the main account number, he must also have access to the new main account. The auxiliary account may be changed to any value without restriction.

It is possible for a user with access to the auxiliary account only, to use this command to change either the main account number (in which case he must have access to the new main account) or the auxiliary account number (to any value desired). By the former action, this user can 'take over' the cost from the original creator of the file, and may remove the original creator's owner access as well. By the latter, he can transfer his own owner access to someone else, by changing the auxiliary account number to an account that someone else has access to.

## Examples:

\*ACCOUNT NewDump 25

changes the account number of NewDump to 25. To do this, the user must have access to both the original account, and also account 25.

\*ACCOUNT DumpProg (37)

If user JOE had access to account 37, then this command would give Joe (as well as this user) owner access to the file DumpProg. This could be useful for several users all working on the same set of files.

\*ACCOUNT New\* 25 (37)

This command changes all items beginning with the letters NEW to account number 25, and auxiliary account 37 (as above).

## Likely Errors:

**Insufficient access**                      **Error 189 (BD)**  
Only an owner can change the account of a file.

**Account nn bankrupt**                  **Error 198 (C6)**  
There must be sufficient credit in the new main account.

## Associated Keywords:

\*CREDIT, \*DEBIT, \*STATEMENT

## Compatibility Notes:

This command does not exist in Acorn systems, which do not support space accounting. Acorn systems use a different system to determine file ownership.

\* ACCOUNT \* 700



changes all to  
700.

**Syntax:** See the BBC Microcomputer User Guide, Section 33.

## Description:

These commands are detailed in the BBC Micro User Guide in the section containing descriptions of all keywords alphabetically. The effects of using the BPUT#, BGET#, INPUT# and PRINT# commands when the access to a file is limited, is explained under the \*ACCESS command (see this Section).

*Note that the use of BPUT#, BGET#, INPUT# and PRINT# is very slow over the network from a BBC Microcomputer.* The reason for this is that BASIC (and some other languages) sends bytes one at a time, requiring a complete network transaction (about 50 bytes sent in total plus File Server overheads) for each single byte of useful information.

If there is much string fetching and putting to be done, the use of the OSGBPBP machine code call (See Chapter 7) is recommended. An easy way to do this is to run the transient program \*PUTGET before beginning the session; this plants code to convert single byte filing operations into the appropriate block operation: see the Section on \*PUTGET in this chapter.

Users may alternatively wish to use their own call to the OSGBPBP routine, especially if they find the restrictions on the use of PUTGET make it unsuitable:

## Procedure to call OSGBPBP from BASIC

At the head of the program allocate workspace,

```
DIM gb% 12           this is space for the arguments to OSGBPBP
```

and space for a buffer if required, e.g.,

```
DIM BUFF% 500       (for writing, this could alternatively be a string)
```

followed by, for example,

```
PROCgbpb (3, CHAN%, BUFF%, 100, 2000)
```

with this definition after the end of the main program,

```
DEFPROCgbpb (A%, channel%, buffer%, length%, offset%)
  LOCAL X%, Y%
  X%=gb%
  Y%=gb% DIV 256
  ?X%=channel%
  X%!1=buffer%
  X%!5=length%
  X%!9=offset%
  CALL &FFD1
ENDPROC
```

This example would load 100 bytes from offset 2000 in the file whose channel number is channel%, to the location given by the value of BUFF%

Other values of A% will have the following effects:

A%=1 Write bytes from buffer% to file, at offset% bytes from start of file  
A%=2 Write bytes from buffer% to file, at current value of PTR#

A%=3 Read bytes from file to buffer%, at offset% bytes from start of file  
A%=4 Read bytes from file to buffer%, at current value of PTR#  
A%=5 to A%=8 other functions (See §10)

## Likely Errors:

**xxxx is not a file**

**Error 181 (B5)**

A directory may be opened using OPENIN (only), but an attempt to use BPUT# or BGET#, or to set or read PTR# or EXT# will give this error.

**Insufficient Access**

**Error 189 (BD)**

An attempt to read a file to which the user has only write access, or to write to a file to which the user has only read access.

**File not open for update**

**Error 193 (C1)**

An attempt to write to a file that has been opened with OPENIN, or to write to a file with access W only with PTR# not equal to EXT#.

**Channel**

**Error 222 (DE)**

Most likely because the file has not yet been opened, because BREAK has been pressed, or because the File Server has been restarted or the discs have been changed. Log on again (after BREAK or restart only !), and use OPENIN, OPENUP or OPENOUT as appropriate to open the file before using one of these operations.

**EOF**

**Error 223 (DF)**

After an attempt to read data beyond the end of the file.

## Compatibility Notes:

All these keywords are supported by Acorn systems. There are some small differences in the detailed interpretation. The use of \*PUTGET is also recommended with Acorn File Servers.

---

**Syntax:** \*BUILD <file specifier>

## Action with Wild Cards in the File Name:

Wild cards prohibited.

## Description:

This program creates a new file of the name <file specifier>, and then prompts for keyboard input, which is sent directly to the file. To end the file press the <Escape> key. The file will be closed, and control returned to the current language.

A common use of this program is to create !BOOT (see under \*OPT4, this Section) or other files which are going to be used with \*EXEC, although any text file may be entered using \*BUILD.

This program opens a new file using the OPENOUT call, and then writes to the file using the multiple byte transfer operation OSGBP. It will therefore run much faster than, for example, the version of \*BUILD contained in the DFS ROM. On a machine equipped with DFS, this program should be run by typing \*/BUILD (see under \*RUN for details).

## Examples:

```
*BUILD !BOOT

0001..... (type
0002.....  your
0003.....  text
0004.....  here)
0005      <Escape>
```

## Likely Errors:

There are no errors specific to this program, but it opens the file using OPENOUT, so it can cause the same errors as OPENOUT (see this Section).

## Compatibility Notes:

Supported on Acorn systems.

**Syntax: \*BYE****Description:**

This command logs off from the currently selected File Server. The user's name and machine number are cleared from the current user list within the File Server. In addition the user's currently selected directory (CSD), user root directory (URD) and Library directory are de-selected, and any open files are closed.

It is recommended that all users use the \*BYE command at the end of a session otherwise someone else using their station later on will have access to all this user's files and accounts. This is especially important for system users or others with access to special information.

Some applications programs may log on to multiple File Servers, and use the appropriate Osword calls to select between them. In this case \*BYE will only log off from the most recently selected of these.

If <Ctrl B> is active (as you are printing) \*BYE will not close the printing job. <Ctrl C> must be used.

**Likely Errors:**

Who are you ?                      Error 191 (BF)

If the user was not logged on. This error will also be produced if a filing operation is attempted after the \*BYE command.

**Associated Keywords:**

\*LOGOFF

**Compatibility Notes:**

Supported by Acorn systems.

**Syntax:** **\*CAT** [<directory specifier>] | **\*.** [<directory specifier>]

## Action with Wild Cards in the Directory Name:

Occurs on first match only (alphabetically).

## Description:

Although all File Server commands may be abbreviated with a dot at the end, it is worth noting that the minimum abbreviation of this command is **\***.

This command causes a list to be printed of the contents of the directory <directory specifier>, or of the currently selected directory (CSD) if the specifier is omitted. The form of the list is:

<directory name>(<seq. no.>)	<access status>
<currently selected disc>	Option <option number>
Dir. <currently selected directory>	Lib. <currently selected library>
{<file name> <access letters>}	

The <directory name> is that of the directory being displayed, with a sequence number <seq. no.>, which is incremented every time a change is made to the directory. The <access status> is either *owner* or *public*, depending on whether the user has access to the account numbers of the directory. The name of the currently selected disc (see under **\*SDISC**) is also displayed.

The <option number> is that set up by the user using the **\*OPT4** command (see this Section). The <disc name> is the name of the currently selected disc. The currently selected library (usually \$.LIBRARY, but see under the **\*LIB** command) will appear after **Lib.** and the currently selected directory after **Dir.**

There then follows a list of the files in the directory, in alphabetical order, with their access status (see under the **\*ACCESS** command). Some of the access letters (either before or after the / character) will be upper case, and the others will be lower case. The upper case letters indicate the access that this user actually has to this item. For example, access letters **WR/r** means that the user is an owner of the file, and can read it or write to it. If the access were **wr/R**, the user would have public access only to the file, and would be able to read it only. If the owner has set an item to access **P**, then it will appear only as **...Private** to a non-owner.

## Examples:

**\*.**

lists the currently selected directory, and will produce output similar to:

John (038)	Owner
MAIN-1	Option 03 (Exec)
Dir. John	Lib. LIBRARY
ABC	WR/r
File23	WR/r
...	
...	

**\*.** PROGS

lists the contents of directory PROGS, which is itself in the currently selected directory, and will produce output similar to:

PROGS (038)	Owner
MAIN-1	Option 03 (Exec)
Dir. John	Lib. LIBRARY

ADDER	WR/r	ScreenDump WR/r
-------	------	-----------------

...  
...

\*CAT \$.FRED.DATA

lists the contents of directory DATA, itself in directory FRED, which is in the root of the currently selected disc.

\*.:MAIN2.JOHN.PROGRAMS

lists directory \$.JOHN.PROGRAMS on disc MAIN2 (which may or may not be the currently selected disc).

### Likely Errors:

**xxxx is not a directory**                      **Error 190 (BE)**  
If the specifier after the command is the name of a file.

**xxxx not found**                              **Error 214 (D6)**  
If the directory specified cannot be found.

### Associated Keywords:

\*CATALL, \*EX, \*INFO, SIZER

### Compatibility Notes:

Supported by Acorn systems but the access letters all appear in upper case on Acorn systems.

**Syntax:** \*CATALL [<directory specifier>]

## Action with Wild Cards in the Directory Name:

Occurs on first match only (alphabetically).

## Description:

\*CATALL produces a catalogue of the specified directory, and of all sub-directories within it. If the specifier is omitted, this catalogue is given for the currently selected directory. The display is of the form:

```
<file name>
<file name>
  Directory <directory name>
    <file name>
    <file name>
    Directory <directory name>
      <file name>
      .
      .
      .
    <file name>
    Directory <directory name>
      <file name>
      .
      .
      .
```

The file and directory names in the left-most column are those in the specified directory. Each time a sub-directory is encountered, its contents are listed, indented four spaces to the right. In this way a complete tree of directories is displayed.

## Examples:

\*CATALL

```
!BOOT
!MAIL
BTWiring
contnts
Ecommands
```

```
Directory Letters
  ADDRESSES
  Base
```

```
Directory Bloggs
  12-7-85
  13-5-85
  cougar3
```

```
Directory Customers
  Alpha
  Bravo
```

Charlie  
Pers  
RR

PokerBoot

Directory ReturnNote  
    general  
    Grommet  
StEdsEP  
ur2

Files !BOOT, !MAIL, down to PokerBoot, StEdsEP, ur2 are in the CSD. Letters, ReturnNote are sub-directories of the CSD, and Bloggs, Customers are sub-directories of Letters.

### **Likely Errors:**

**xxxx is not a directory                      Error 190 (BE)**  
If the specifier after the command is the name of a file.

**xxxx not found                              Error 214 (D6)**  
If the specified directory cannot be found.

### **Associated Keywords:**

\*CAT, \*EX, \*INFO, SIZER

### **Compatibility Notes:**

Supported by Acorn systems.

---

**Syntax:** \*CDIR <directory specifier>

### Action with Wild Cards in the Directory Name:

Wild cards prohibited.

### Description:

This command creates a directory of that name. Note that if a directory of that name already exists, no action will be taken (and no error message will be produced). The user must have access to the main or auxiliary account of the directory in which he is creating this new directory: but he may then change the account of the new directory to any other to which he has access.

### Examples:

\*CDIR PROGRAMS

creates a directory called PROGRAMS in the currently selected directory

\*CDIR \$.PROJECT.NEWDATA

creates a directory NEWDATA in the directory \$.PROJECT. The directory PROJECT must already exist, and the user must have access to the account or the auxiliary account of the directory PROJECT.

### Likely Errors:

#### Bad Wildcard

#### Error 204 (CC)

An attempt to use a wild card character \* or # in the new directory name.

#### Account nn bankrupt

#### Error 198 (C6)

There must be sufficient credit in the account of the directory in which the user is creating this new directory.

#### Insufficient access

#### Error 189 (BD)

If the user does not have access to the main or auxiliary account of the directory in which the user is creating this new directory.

#### xxxx is not a directory

#### Error 190 (BE)

If a file (not a directory) of that name already exists.

#### xxxx not found

#### Error 214 (D6)

If the directory into which the new one is being created is not found. If, in the second example, the directory \$.PROJECT did not exist, the error PROJECT not found would occur.

### Associated Keywords:

\*ACCOUNT, \*DEFACCESS

### Compatibility Notes:

Supported by Acorn systems except that Acorn systems do not support accounts, but use a different system to determine ownership. Note that Acorn File Servers will give an error after an attempt to create a directory if a non-empty (or locked) one of that name already exists. For compatibility, users' programs that use

\*CDIR should use the OSFILE call with A=5, which will check for the existence of the item (will return A=0 if there was no file or directory or file with the name specified, A=1 if a file of that name was found, A=2 if a directory was found). The important feature of this call is that the library will not be searched for a directory (though it will be for a file).

---

## **Syntax: \*CLOSE**

### **Description:**

This program closes all currently open files, but not any open directories. It is exactly equivalent to the BASIC command **CLOSE#0**, but can be run from any language or other system.

\*CLOSE runs in Page 0 of the BBC Microcomputer, and so will not corrupt other transient programs.

### **Examples:**

\*CLOSE

### **Likely Errors:**

There are no errors specific to this command.

### **Associated Keywords:**

\*BYE

### **Compatibility Notes:**

Supported by Acorn systems.

**Syntax: CHAIN "COPIER"****Action with Wild Cards in the File Name:**

Occurs on first match only (alphabetically).

**Description:**

This program is a general purpose copying utility, which will copy files of any size between any two filing systems, enter the names of the two systems ( e.g. DISC and NET) after the appropriate prompts. To copy between two directories, enter **\*DIR <directory pathname>** after each, and similarly to copy between File Servers type **\*I AM <File Server station number><user id.>** after each.

The program will then prompt for the file name, and for the new name required in the destination filing system. Typing **<Return>** after the latter will give the same name to both. The system will then continue to prompt for files to be copied until **<Escape>** is pressed.

COPIER will **\*LOAD** and then **\*SAVE** the file if it is sufficiently small, otherwise it will call **OPENOUT** to create the new file, and will then read blocks of data from the source and write them to the destination, using **OSGBPB** calls.

\* commands may be typed after either of these prompts, but note that, where a command applies specifically to one of the filing systems, it will apply to the source system if typed after **File name**, and to the destination system if typed after **New name**.

If a file is to be *moved* within one disc on a File Server, note that the **\*RENAME** command (see this Section) will do this, and it is not necessary to use **COPIER**.

Where it is necessary to copy multiple files between File Servers, the utility **MULTICOPY** is provided. **MULTICOPY** will copy entire directory trees automatically.

**Examples:**

After **CHAIN "COPIER"**, the commands:

Source filing system: **\*DISC**  
Dest. filing system: **\*NET**

will copy from floppy disc to network File Server. If it is necessary to select a directory (for example), this can be done with the **\*DIR** command after the **New name** prompt.

Source filing system: **\*I AM 253 FRED**  
Dest. filing system: **\*I AM 254 FRED**

will copy selected files between File Servers, probably on behalf of user **FRED**. After the file name prompts, it is possible to enter a full file specifier (e.g. **\$.JOHN.BBCPROGS.THING**). To save files the user must of course be an owner of the destination directory.

Source filing system: **\*DIR \$.JOHN**  
Dest. filing system: **\*DIR \$.FRED**

This will copy selected files from **\$.JOHN** to **\$.FRED** on the same File Server. This is useful if **FRED** needs his own copies of the files (if for example he was likely to change his copy). If this is not the case, use **\*RENAME**.

## **Likely Errors:**

This program will respond **Come again ?** if the file is not found. Otherwise the errors produced by \*LOAD, \*SAVE, OPENIN and OPENOUT can occur when running this program.

## **Associated Keywords:**

\*MULTICOPY

## **Compatibility Notes:**

Supported by Acorn systems.

---

**Syntax: \*cv****Description:**

This program displays the station numbers of the currently selected File Server and Printer Server, the user's own station number, and the version of OSARGS in use. If there are multiple networks joined by bridges, then the network number will be the value that would be returned in A by the OSARGS call with Y=0 and A=2. This number depends on the local NFS version number:

OSARGS version no. 2 - NFS 3.34

OSARGS version no. 1 - NFS 3.6 or Advanced NFS

**Examples:**

\*CV

FS number 254  
PS number 235  
You are 001.005  
OSARGS ver 001

**Likely Errors:**

The message **RxCB ?**

will be displayed if no receive control block is available in the BBC Microcomputer.

There are no other errors specific to this program.

**Associated Keywords:**

\*FSLIST, \*PSLIST

**Compatibility Notes:**

Supported by Acorn systems.

---

**Syntax:** \*DEFACCESS [<directory specifier>] <access string>

## Action with Wild Cards in the Directory Name:

Occurs on first match only (alphabetically).

## Description:

This command applies to the directory specified, or if the specifier is omitted to the CSD. The string will be applied as the default access string to any new files created within this directory, or with any subsequently new files created in this directory, or within any subsequently created subdirectory (unless DEFACCESS is used again to apply to that subdirectory).

For a list of the possible access letters, see the \*ACCESS command. The default access for a directory will be listed by the \*EX or \*INFO commands (see this Section).

The user must be an owner of the specified directory.

## Examples:

```
*DEFACCESS WR/R
```

This sets the default access of the files in the currently selected directory to read and write access for owners and read access only for non-owners. The root directory \$ on a new File Server disc has this default access, so directories in \$ will have this default unless it is explicitly changed.

```
*DEFACCESS $.SECRET PWR/
```

will cause subsequently created files in the directory \$.SECRET to have access letters **PWR/** and subsequently created sub-directories to have access **PD/**.

## Likely Errors:

### FS Unusual Error 70 (46)

An attempt has been made to set an illegal attribute (e.g. D as part of the default access).

### Insufficient access                      Error 189 (BD)

Only an owner can change the default access status of a directory.

### xxxx is not a directory                  Error 190 (BE)

If the specifier after the command is the name of a file.

### Bad attribute                              Error 207 (CF)

Attempt to use letters other than P D W L R.

### xxxx not found                            Error 214 (D6)

If the directory specified cannot be found. If, in the second example, the directory \$.SECRET did not exist, the error **SECRET not found** would occur.

## Associated Keywords:

\*ACCOUNT

## Compatibility Notes:

This command is not supported in Acorn systems. The defaults are **WR/** for files, and **DL/** for directories on Acorn File Servers.

**Syntax:** \*DELETE <general specifier>

## Action with Wild Cards in the File Name:

Occurs on every match. The command \*ENABLE will be needed before a wild card delete if the system manager has set the appropriate option. Note \*DELETE must be used on ANFS.

## Description:

This command deletes the specified file. The user must be an owner of the file, and the file must not be locked (access letter L).

When used with a wild card specifier, this command will delete *every* unlocked item that matches, within the specified, or currently selected, directory. The system manager can set an option to require the command \*ENABLE before any wild card delete operation. If it is required to delete a tree of directories and sub-directories, the program ERAQ (see this Section) should be used.

It is only possible to delete a directory if it contains no entries.

## Examples:

```
*DELETE VATPROG
```

deletes the file (or directory) VATPROG in the currently selected directory.

```
*ENABLE  
*DELETE *.*
```

finds the first directory (alphabetically) in the currently selected directory, and deletes all items within this first directory (except ones where the attempt would cause an error). This is standard action with wild card specifiers, and is explained fully in Section 6.3 at the beginning of this chapter.

```
*DE. $.JOHN.MYPROG
```

Deletes the file MYPROG in directory \$.JOHN. The user must be an owner of (i.e. have access to the main or auxiliary account of) the file MYPROG.

There is also a command \*DISABLE, which has the opposite effect of \*ENABLE (both are described fully in this Section).

## Likely Errors:

**Directory not empty**                      **Error 180(B4)**

Caused by an attempt to delete a directory that still contains some entries.

**Insufficient access**                      **Error 189 (BD)**

Caused if the user is not an owner (i.e. does not have access to the main or auxiliary account) of the item.

**Not ENABLED**                              **Error 189 (BD)**

If the system manager has set the \*ENABLE required option for this user, the Not ENABLEd error will be given.

**Already opened by xxxx**                      **Error 194 (C2)**

If another user has this file open for reading or writing, then it cannot be deleted until it has been closed.

**Entry locked****Error 195 (C4)**

It will be necessary to use the \*ACCESS command to unlock the file before it can be deleted.

**Bad wildcard****Error 204 (CC)**

Use of a wildcard on an ANFS without using \*DELETE

**Associated Keywords:**

\*ENABLE, ERAQ

**Compatibility Notes:**

Supported by Acorn systems, but note that wild cards are not allowed.

---

**Syntax:** \*DIR [<directory specifier>]

## Action with Wild Cards in the Directory Name:

Occurs on first match (alphabetically) only.

## Description:

This command changes the currently selected directory (CSD) to the one specified. If it is used without a specifier, it will select the user root directory (URD) as the CSD.

There are several special characters that can be used to specify directories: & can be used to represent the URD in pathnames, ^ refers to one level up the directory tree each time it is used, and @ specifies the CSD. The last option is useful if a program asks for a directory name to add to \*DIR, as simply pressing <Return> would go back to your URD.

When a directory is selected by this command, then it is deemed to be open by the File Server. The user root directory (URD) and library directory are opened at log-on, and remain open throughout the session. Hence the number of channels available for other filing operations is reduced by two, from the maximum of eight, if the \*DIR command has not been used, and by a further one if \*DIR has been used to select a different directory as the CSD. In addition, \*DIR opens the new directory before it closes the old one (if appropriate), giving an instantaneous total of four directories open.

## Examples:

DIR PROGRAMS

searches the currently selected directory for a directory called PROGRAMS, and makes that the new currently selected directory.

\*DIR \$.FRED.LETTERS

selects the directory LETTERS in directory \$.FRED as the CSD.

\*DIR

selects the user root directory as CSD.

\*DIR &.^.^

selects the directory two levels up the tree from your URD as the new CSD.

## Likely Errors:

xxxx not found                      **Error 214 (D6)**  
If the specified directory could not be found.

Too many files open                **Error 192 (C0)**  
If several files are open for random access, it is possible that there may not be a channel left for the CSD.

xxxx is not a directory            **Error 190 (BE)**  
Caused by an attempt to specify a file as the CSD.

## **Associated Keywords:**

\*DISC

## **Compatibility Notes:**

Supported by Acorn systems, but wild cards and the character ^ are not allowed.

---

**Syntax: \*DISABLE || \*DISABLE SAVES || \*DISABLE LIBRARY**

## **Description:**

This command is the exact opposite of the \*ENABLE command (see this Section).

**\*DISABLE** without a parameter prevents the use of \*DELETE with a wild-card thereafter, until \*ENABLE is typed. This command may be run automatically at log-on, depending upon an option set for a user by the system manager in the password file.

**\*DISABLE SAVES** prevents a file of less than 16 bytes in length being saved. This command may be run automatically at log-on, depending upon an option set for a user in the password file, by the system manager.

**\*DISABLE LIBRARY** reduces the library searching facility provided in SJ Research File Servers, to a level equivalent to that in Acorn systems. That is to say, the library is searched only for \*RUN or \* commands, and not otherwise.

The effect of these three commands may be reversed by use of the corresponding \*ENABLE command. Their effect otherwise persists until the user logs off.

## **Examples:**

```
*DISABLE  
*DISABLE SAVES  
*DISABLE LIBRARY
```

## **Likely Errors:**

There are no errors specific to this command. If the word after \*DISABLE is not recognised, then the effect will be that of \*DISABLE (without an argument).

## **Associated Keywords:**

\*ENABLE

## **Compatibility Notes:**

Not supported by Acorn systems.

---

## Syntax: \*DISCS

### Description:

This program produces a list of all the discs present in the system, in order of their drive numbers. The free space on each disc is also given. It will not list the tape drive; although this, if fitted, is treated as a virtual disc called %TAPE on the File Server.

This command is identical to \*FREE (see this Section), but is provided for compatibility with Acorn libraries. The system manager may delete it if it is not required.

### Examples:

\*DISCS

will produce the following response from the system (for example):

Drive	Name	Bytes free,	used
0	DISC1	560k	481k
1	DISC55	1098k	113k

### Likely Errors:

There are no errors specific to this program.

### Associated Keywords:

\*FREE

### Compatibility Notes:

Supported by Acorn systems.

---

**Syntax:** **\*DUMP** <file specifier> [<offset>]

## Action with Wild Cards in the File Name:

Occurs on first match (alphabetically).

## Description:

This program opens the specified file, and prints it as a hexadecimal dump on the screen of the computer. The output will start at the address <offset> (in hexadecimal) if one is specified.

The output consists of lines of the form:

```
aaaaaa nn nn nn nn nn nn nn  cccccccc
```

where **aaaaaa** is the offset from the beginning of the file of the left-most byte displayed, the **nn** are the hexadecimal values of 8 bytes of the file, and the **c** are the same 8 bytes in character form. If there exists no character form (if for example the byte value is less than 32 or more than 126), then the character printed will be a dot.

This program uses the multiple byte transfer OSGBPB call, and so will run considerably faster than for example the version of **\*DUMP** that is contained in the DFS ROM. If DFS is fitted to the computer, use **\*/DUMP** to be sure of running the network version. (See under **\*RUN** for full details of \*).

## Examples:

```
*DUMP FILE1
```

will dump FILE1 to the BBC Microcomputer screen. A printed copy could be made at the same time by typing <Ctrl-B> before the command, to turn on the printer, and <Ctrl-C> at the end to turn it off. See Sections 5.5 and 6.5 about printing through the network.

```
*DUMP LOGFILE 1A000
```

will dump LOGFILE, starting 1A000 (decimal 106496) bytes from the start of the file.

## Likely Errors:

There are no errors specific to this program. However, it performs an OPENIN call, and so can cause all the same errors that the OPENIN would.

## Associated Keywords:

**\*TYPE**

## Compatibility Notes:

Supported by Acorn systems.

---

**Syntax: \*ENABLE || \*ENABLE SAVES || \*ENABLE LIBRARY**

## Description:

Used without a parameter, this command allows the \*DELETE command to be used with wild cards, i.e. with a file specifier after \*DELETE containing the characters \*, . or #.

The system manager can set the 'Permanently ENABLEd' option in the password file for any user, to enable that user to perform wild card delete operations all the time. If this option has been set, then the \*ENABLE command is redundant, unless \*DISABLE (see this Section) has been used.

If a user attempts to use a wild card delete without either having typed \*ENABLE or having the 'Permanently ENABLEd' option set, then the error BD will occur, and the message **Not ENABLEd** will be given.

If \*ENABLE is followed by SAVES, this permits the user to save files of less than 16 bytes in length (the system manager can set an option to prevent any user from saving these short files). \*ENABLE SAVES over-rides this option, and allows any user to save short files -- its effect lasts until log-off, or until a \*DISABLE SAVES command.

The reason for not allowing files of shorter than 16 bytes to be saved, is to prevent users accidentally destroying BASIC programs by saving over them after pressing <Break>, without first typing **OLD**. Users who are likely to do this are warned against the indiscriminate use of \*ENABLE SAVES.

The command \*ENABLE LIBRARY causes the currently selected library to be searched for any \*, \*RUN, load or 'open file for input' command. This is the default setting for SJ Research File Servers (but not for Acorn FSS), but it may have been turned off with \*DISABLE LIBRARY.

The effect of these \*ENABLE commands persists until the user logs off, or until he types the corresponding \*DISABLE (see this section).

## Examples:

\*ENABLE  
\*ENABLE SAVES  
\*ENABLE LIBRARY

## Likely Errors:

There are no errors specific to this command. If the word after \*ENABLE is not recognised, then the effect will be the same as for \*ENABLE without an argument.

## Associated Keywords:

\*DISABLE

## Compatibility Notes:

Not supported by Acorn systems.

---

**Syntax: CHAIN "ERAQ"****Action with Wild Cards in the Directory Name:**

Occurs on first match (alphabetically).

**Description:**

This program deletes part or all of an entire directory tree.

The first question will be:

Do you want to OK the files before deletion ?

Typing **Y** will cause the program to prompt for **Y** or **N** after displaying the name of each file. Typing **N** after the first question will cause the entire directory tree to be deleted.

The program will then prompt:

Full path name ?

The user should enter the full name of the directory tree (starting at the root), that he wants to delete. Entering **\$** on its own would attempt to clear the entire disc (possible only for the system manager to do !). For a user to clear part or all of his files, he should enter the full path name, starting at **\$**.

The name of each file or directory will then be displayed, followed by the prompt for **Y** or **N** if the answer to the initial question was **Y**.

ERAQ will stop if a file or directory is found that the user does not own (i.e. have access to its account). When it stops, the CSD will be changed to the last directory visited by the program.

Note that this program will delete files **even if they are locked** users are warned to take care, especially if answering **N** to the initial question.

**Examples:**

CHAIN "ERAQ"

Do you wish to OK the files before deletion ? N

Full path name ? \$.JOHN.PROGS

\$.JOHN.PROGS.Main1  
\$.JOHN.PROGS.Silly  
\$.JOHN.PROGS.BBC.EDITOR  
\$.JOHN.PROGS.BBC.eraser

.  
.  
.

\$.JOHN.PROGS

and so on, including all sub-directories contained within \$.JOHN.PROGS.

## **Likely Errors:**

There are no errors specific to this program, but ERAQ uses the \*DELETE call, and so can cause the errors associated with \*DELETE. There is also the possibility of getting the BASIC error String too long if the directory structure is very deep.

## **Associated Keywords:**

\*DELETE

## **Compatibility Notes:**

Supported by Acorn systems, but can be used only by a system privileged user.

**Syntax: \*EX [<directory specifier>]****Action with Wild Cards in the Directory Name:**

Occurs on first match (alphabetically).

**Description:**

This command gives information similar to that produced by \*INFO, but for every file in the specified directory. If no directory is specified, then the currently selected directory will be displayed.

The information given is a header identical to that given by \*CAT, followed by a line of the form below for each file:

<file name> <load addr.><execute addr.><length><access><date1><date2><time> xx(yy)

or

<directory name> No of entries=nn Default=xx/xx<access><date1><date2><time> xx(yy)

or

<job name> <User Id> at Stn. sss <length> <access> <printer> <date1><time> xx(yy)

or

...Private

where

<load addr.>	is the address (in hexadecimal) in which the file would be loaded by the *LOAD or *RUN commands.
<execute addr.>	is the address (in hexadecimal) at which execution would begin in a *RUN command.
<length>	gives the true length of the file in bytes (in hexadecimal).
<access>	is the set of access letters for that item, as listed under the *ACCESS command.
<date1>	is the date that the item was originally created,
<date2>	
<time>	are the date and time that the item was last changed, i.e. written to or saved over in the case of a file, or contents changed in the case of a directory. If the date is today's, then <b>today</b> will appear in the <date1> and <date2> fields.
<User Id>	is the user who submitted the job for printing.
<printer>	is the logical printer selected for the job.
xx	is the account number associated with the file (see under *ACCOUNT).
yy	is the auxiliary account number associated with the file.
sss	is the station number from which the print job was submitted.
Default=xx/xx	gives the default access letters for a directory (i.e. the access status given to any file saved in it, until this is changed using the *ACCESS command). The default setting may be changed using *DEFACCESS.

The word ...Private only is given to a non-owner of an item, if the access of the item is P.

## Examples:

**\*EX**

examines the contents of the currently selected directory, and will give an output similar to:

```
NEXTISS      (027)  Owner
MAIN-IV      Option 03 (Exec)
Dir. NEXTISS  Lib. LIBRARY

CONT          00000000      FFFFFFFF  0038A6  WR/r      11jul85
15jul85 12:55 F0 (00)
GT-Eg         00000800      00008023  00017C  WR/r      11jul85
15jul85 14:30 F0 (81)
PUTGET        00000000      FFFFFFFF  0013B5  WR/r      08jun85
11jun85 18:55 F0 (00)
temp          00000000      FFFF3200  002231  WR/r      15jul85
15jul85 14:28 F0 (00)
wombat        FFFFFFFF      FFFFFFFF  000000  WR/r      17jul85
today 22:32 F1 (00)
```

**\*EX \$**

examines the system root directory on the disc. This will list out all the users' root directories and any other directories and files saved in directory \$.

**\*EX Progs**

will examine the contents of the sub-directory Progs, which is in the currently selected directory.

**\*EX \$.JOHN.BBC-PROGS.OLD**

will look at directory OLD which is in BBC-PROGS in \$.JOHN

## Known Bugs:

There is a bug in NFS 3.6, where the first letters \*EX of a longer command beginning with \*EX... will be stripped off before passing the name on to the File Server. For example, \*EXAMINE will attempt to \*RUN a file called AMINE. It is wise to use \*RUN or \*/ with any file beginning with the letters EX.

## Likely Errors:

**xxxx is not a directory**                      **Error 190 (BE)**  
If the specifier after the command is the name of a file.

**xxxx Not Found**                              **Error 214 (D6)**  
If the directory specified cannot be found.

## Associated Keywords:

\*CAT, \*CATALL, \*INFO, SIZER

## Compatibility Notes:

Similar on Acorn systems, but without account information, or the date of original creation field. Print spooling is also not available on Acorn systems. A number known as the SIN will appear in the space in which an SJ Research File Server would display the account numbers.

---

**Syntax:** \*EXEC <file specifier>

## Action with Wild Cards in the File Name:

Occurs on first match (alphabetically).

## Description:

This command opens the file specified for input, and then executes text from it as though it had been entered from the keyboard of the BBC Microcomputer. The file is closed automatically at the end of the text in it, but will remain open if an error occurs in the middle of the text sequence.

The \*EXEC command is useful for performing sequences of commands repeatedly. Using either the \*BUILD utility, or the \*SPOOL command (see this Section), or a word processor, the sequence of commands can be built up in a file.

It is also possible to convert a text file into a BASIC program with this command. Normally, as a BASIC program is typed in at the keyboard, it is converted into a condensed *token* form by the language system. This is then the form in which it is saved in a file, or re-loaded. Sometimes, however, it can be more convenient to create a program using a word processing package, and then submit it to BASIC. This can be done by saving the text file, and then typing \*EXEC followed by the file name.

Note that the \*EXEC command opens the file and then reads text, using the BGET call. This is very slow on the network, so it is recommended that the user runs the utility PUTGET (see this Section) before this command, if the file to be \*EXECed is very long.

## Examples:

\*EXEC COMMANDS

will perform the commands contained in the file COMMANDS. If COMMANDS contains the text form of a BASIC program, this will be entered as though it were keyboard input to the currently selected language.

## Likely Errors:

### File not found

### Error 214 (D6)

If the file does not exist. Note that the system replies **File not found** (normally the File Server replies <file name> not found).

There are no other errors specific to this command, but it calls OPENIN (see this Section) and so can cause the same errors as that command.

## Associated Keywords:

\*BUILD

## Compatibility Notes::

Supported by Acorn systems.

## **Syntax: \*FLUSH**

### **Description:**

This command causes the contents of any printout to be flushed. It will be found useful if a user's program has generated large quantities of spurious output.

Any printout, waiting to be printed, in the %PRINTQ directory belonging to the user will also be cleared. To determine whether the files in the print queue belong to the user typing \*FLUSH, the station number of the computer and the user identifier must be the same as the user issuing the command.

To selectively remove files from the print queue \*DELETE should be used.

Note that printers themselves often have an internal buffer, which means that they could carry on printing for some pages after a \*FLUSH command. To clear a printer's internal buffer, it will be necessary to turn the printer off and on.

### **Likely Errors:**

There are no errors specific to this command

### **Associated Keywords:**

\*PGO, \*PSTOP

### **Compatibility Notes:**

Not supported by Acorn systems.

---

**Syntax: \*FREE****Description:**

This program produces a list of all the discs present in the system, in order of their drive numbers. The free space on each disc is also given. If your File Server is fitted with a tape drive, this will be treated as a virtual disc called %TAPE, but will not be listed by \*FREE.

**Examples:**

\*FREE

will produce the following response from the system (for example):

Drive	Name	Bytes free	Bytes used
0	DISC1	522k	21086k
1	Main-05	3496k	18112k

**Likely Errors:**

There are no errors specific to this program.

**Associated Keywords:**

\*DISCS

**Compatibility Notes:**

Supported by Acorn systems.

**Syntax:** **\*FS** [<network number>.<station number>]

### **Description:**

**\*FS** is only available on the Advanced version of the N.F.S. ROM (ANFS); see under **\*HELP** for how to find out which ROMs are fitted to your station.

This command allows you to change the File Server station number stored in the BBC Microcomputer, while it stores the handles you were using on the File Server you have just left. Handles can be stored for up to five File Servers at once, although this depends on how many files you have open.

Thus you can swop between File Servers which you are logged on to, without having to repeatedly log on and lose your CSD and open files.

### **Likely Errors:**

If a station which is not a File Server is selected, the message **Station nn not listening** will be given when any filing operations are attempted.

### **Associated Keywords:**

**\*FSLIST**, **\*I AM**

### **Compatibility Notes:**

Supported by Acorn systems for stations with the ANFS ROM.

---

## Syntax: \*FSLIST

### Description:

\*FSLIST displays a list of all *active* File Servers available in an installation. A File Server will not be displayed in this list if it is in Configuration or Utility Mode, or if it has crashed.

Note that, in contrast to this, \*STATIONS will display File Servers, even if they are inactive.

If the Econet installation comprises multiple networks, \*FSLIST will also display File Servers on other networks, preceded by their network number (e.g. File Server 253 on network 2 will be displayed as **002.253**)

### Examples:

\*FSLIST

File Servers/Type

235	SJ Research File Server ver M.97/HDFS
064.127	SJ Research File Server ver 0.91/FDFS
064.182	SJ Research File Server ver 0.90/FDFS
064.235	SJ Research File Server ver M.98/MDFS
200	SJ Research File Server ver M.97/HDFS

### Likely Errors:

There are no errors specific to this command.

### Associated Keywords:

\*FS

### Compatibility Notes:

Supported by Acorn systems.

**Syntax:** **\*GNET** followed immediately by a GET statement.

## Description:

This command puts a character into the BBC Microcomputer's buffer, so that it can be read by a program using a GET command (which must follow \*GNET immediately). The value read will be the network number for your station, which will be in the range 0 to 255. This number is required if the network being used contains *bridges* to other networks, in order to access a station on the other side of a bridge.

Do **not** use the BASIC keyword INPUT with \*GNET, since any bytes which are interpreted as control characters will not be stored, but will be interpreted directly by the input routine. Also it is wise to ensure that the piece of program containing the GET statements is absolutely correct, since if BASIC finds an error here, the bytes will be entered into the keyboard buffer, possibly causing strange effects.

With bridges this information is also useful when you wish to log-on to a particular File Server, and you do not know whether or not it is on the local network for your particular station (See **\*I AM**). If the File Server is at station 250 on network 3, you will need to log-on with:

```
*I AM 0.250 DIANA | *I AM 250 DIANA
```

if your station is on network 3, the local network for the File Server. However if your station is on network 2, it is necessary to specify the full station number of the File Server, including the network number, by typing:

```
*I AM 3.250 DIANA
```

The full station number of a File Server on your local network will not permit you to log-on and will give the message **Not Listening**, as if the File Server were not present. The File Server must now be specified as 0.250 as the default network has been changed to the inaccessible network 3 by the previous command.

## Examples:

```
10 *GNET
20 net%=GET
30 IF net%=1 THEN room$="office"
40 IF net%=64 THEN room$="coffee room"
50 PRINT "You are now in the ";room$
```

## Likely Errors:

Note that this command will not work on stations fitted with the N.F.S. ROM version 3.34. Also the BBC Microcomputer screen may do strange things if a mistake occurs between executing \*GNET and GETting the byte.

## Compatibility Notes:

Supported by Acorn systems with appropriate ROMs.

---

**Syntax: \*GO <32 bit address>****Description:**

This command causes a jump to the specified address, which should be in hexadecimal (but with no leading & character required). If a full 32 bit address is specified, then the program looks first at the most significant two bytes (four hex digits) of the address.

If the most significant two bytes of the address are FFFF, then the jump will always occur into the I/O processor (i.e. the BBC Micro itself), to the address given by the least significant two bytes. For example **\*GO FFFF2084** will jump to address 2084 (hex) in the BBC Micro.

If the most significant two bytes of the address have any value other than FFFF, then this command will jump into the second processor if one is present; otherwise into the I/O processor.

There is a **\*GO** command built in to the second processor 'tube' interface, but this does not force a jump to the I/O processor if the more significant half is FFFF. If this feature is required, you must type **\*/GO**

Obviously it is vital that you know what you are doing before executing this command, otherwise the most likely outcome is that the computer will crash.

**Examples:**

**\*GO 3000**

will jump to address 3000 (hex) in the second processor (if any), or in the BBC Micro if no second processor is fitted.

**\*/GO FFFF3000**

will invariably jump to address 3000 (hex) in the BBC Microcomputer itself.

**Likely Errors:**

If the incorrect syntax is given to **\*GO**, the message Syntax: **\*GO <32 bit address>** will be displayed, and error number DC (220) will be given.

**Compatibility Notes:**

Supported by Acorn systems.

**Syntax:** **\*GTIME** followed immediately by 5 GET statements.

### Description:

This command plants 5 characters into the BBC Microcomputer, so that they can be read by a program using five GET commands (which must follow **\*GTIME** immediately).

After the two lines above have been run, the values read will be (in order):

- first byte = day of month (between 1 and 31)
- second byte AND &0F = month (between 1 and 12)
- (second byte AND &F0) DIV 16 = years after 1981 (e.g. 1985 will read as 4)
- third byte = hours from midnight (between 0 and 23)
- fourth byte = minutes (between 0 and 59)
- fifth byte = seconds (between 0 and 59)

Do *not* use BASIC keyword INPUT with **\*GTIME**, since any bytes which are interpreted as 'control' characters will not be stored, but will be interpreted directly by the input routine. Also it is wise to ensure that the piece of program containing the GET statements is absolutely correct, since if BASIC finds an error here, the 5 'time' bytes will be entered into the keyboard buffer, possibly causing strange effects.

### Examples:

```
DIM t% 4
*GTIME
    day%=GET
    monthandyear%=GET
    hours%=GET
    mins%=GET
    secs%=GET

IF hours%<12 THEN am$="in the morning"
IF hours%=0 THEN am$="midnight"
IF hours%=12 THEN am$="noon"
IF hours%>12 THEN am$="in the afternoon"
IF hours%>17 THEN am$="in the evening"
IF hours%>12 THEN hours%=hours%-12
IF hours%=0 THEN hours%=12

PRINT "It is now ";mins% " minutes past ";hours%;" "am$
```

### Likely Errors:

There are no errors specific to this command, but the BBC Microcomputer screen may do strange things if a mistake occurs between executing **\*GTIME** and GETting 5 bytes.

### Associated Keywords:

SETIME

### Compatibility Notes:

Supported by Acorn systems that contain a real-time clock (otherwise the time may be nonsense).

---

**Syntax:** \*GUSER  
          INPUT ""user\$

## Description:

This program plants the name of the user logged on at the current station into the BBC Microcomputer, so that it can be read into a string by the BASIC keyword INPUT -- this must follow \*GUSER immediately. The double quote characters mean that a null string is used as a prompt for the INPUT statement, instead of the usual ?.

User names have a maximum length of 10 characters.

If a BASIC error occurs between \*GUSER and the INPUT statement, then the user name will be entered into the keyboard buffer, and will probably cause BASIC to reply **Mistake**.

## Examples:

```
*GUSER
INPUT ""U$
PRINT "Hallo "U$ ", how are you today ?"
```

## Likely Errors:

There are no errors specific to this command.

## Associated Keywords:

\*I AM, \*PUSER, \*CV, \*USERS

## Compatibility Notes:

Supported by Acorn systems, but note that some Acorn File Servers support user names of more than 10 characters.

## **Syntax: \*HELP**

### **Description:**

This command will list all the ROMs in the BBC Microcomputer which respond to it. This is particularly useful for finding out which version of the N.F.S. ROM your station is fitted with, as this affects some network operations. \*HELP may also be used with a parameter to give further information about some ROMs.

### **Examples:**

\*HELP

6502 TUBE 1.10

NFS 3.60

OS 1.20

### **Likely Errors:**

There are no errors specific to this command.

### **Compatibility Notes:**

Supported by Acorn systems.

---

**Syntax:**        **\*I AM** [<network number>][<File Server number>] <User Id> [<password>]

**\*I AM** [<network number>][<File Server number>] <User Id> :  
                 [<password>]

### Action with Wild Cards:

Allowed in User Id field. Will give first match (alphabetically).

### Description:

This command **logs-on** a user to a File Server -- that is to say, it identifies the user to the File Server, and sets up communication channels between the user's computer and the File Server.

The **\*I AM** command must be the first command given to the File Server, otherwise it will reply with the error message **Who are you?** or **Channel** when any filing operation is attempted through the network.

The File Server will search the password file on each disc in turn (starting with the lowest numbered drive) for the <User Id>, and will check the password quoted by the user against any that he may have set up using the **\*PASS** command previously. The File Server will read from the password file the list of accounts to which the user has access, whether the user has system privilege, and will select a library directory (usually **\$.LIBRARY**, unless otherwise set up by the system manager) for the user. The system will also search the disc on which the user's password file entry was found, for the directory specified as the User Root Directory (URD) for the user in the password file, usually with the same name as the user, and will select this as his Currently Selected Directory (CSD) -- see Section 2.3.

If the <User Id> is not found in the password file of any disc, the user will not be able to log on unless the system manager has set up a *default user*; in which case an attempt to log-on with a user identifier unknown to the system will leave the user logged on as this default user. The system manager will normally have set up some automatic response for the default user, for example to prompt the user to log on again.

The BBC Microcomputer initially assumes that the File Server station number is 254, unless this is otherwise specified. If the required File Server is at another station (note that there may be more than one File Server on an Econet), then type (for example):

```
*I AM 250 FRED
```

to log-on to a File Server at station 250. Any subsequent **\*I AM** command will now assume File Server 250, until another **\*I AM** command specifies a different File Server station number.

If the network contains *bridges* on to other networks, then it is possible to log on to File Servers on these other networks. To do so, the user must specify the *full station number* of the File Server, which will be of the form <network number>.<station number> For example, to log on to the File Server at station 254 on network number 3, type:

```
*I AM 3.254 FRED
```

The user does not need to re-specify the network number for further operations until he wants to select a different network. For example, to log on to the local File Server station 254, (the local network is always number 0) type:

```
*I AM 0.254 FRED
```

The user may conceal his password from prying eyes by typing a space then a colon after his user identifier.



\* I AM JOHN

\*I AM 235 SMITH

\*I AM SYST :

### Likely Errors:

## Not listening

## Error 162 (A2)

Either there exists no File Server at the specified station number, or it has been switched off.

### No clock

### Error 163 (A3)

The user's station is not plugged into the network, or the network clock is not running. See Section 9.4.

## Bad password

### Error 185 (B9)

An attempt to use illegal characters in the password field (e.g. \* # \$ %) will give this error.

**Wrong password**

### Error 187 (BB)

If the user makes an error in entering his password.

**User not known**

### Error 188 (BC)

If there is no default user, this error will be given if the user's name is not known to the system.

**File not found**

### Error 214 (D6)

If the file !BOOT does not exist in either the URD or library. Note that the system replies **File not found** (normally the File Server replies <file name> not found). You are however logged on at this stage despite this error message.

**Associated Keywords:**

\*BYE, \*FS, \*LOGON, \*LOGOFF, \*PASS

### Compatibility Notes:

Supported on Acorn systems.

**Syntax:** \*INFO [<general specifier>]

## Action with Wild Cards in the File Name:

Occurs on first match (alphabetically).

## Description:

This command prints full information about the file (or directory) specified, or the currently selected directory if none specified. The displayed information for the item is:

<file name> <load addr.><execute addr.><length><access><date1><date2><time> xx(yy)

or

<directory name> No of entries=nn Default=xx/xx<access><date1><date2><time> xx(yy)

or

<job name> <User Id> at Stn. sss <length> <access> <printer> <date1><time> xx(yy)

or

...Private

where

<load addr.>	is the address (in hexadecimal) at which the file would be loaded by the *LOAD or *RUN commands
<execute addr.>	is the address (in hexadecimal) at which execution would begin in a *RUN command
<length>	gives the true length of the file in bytes (in hexadecimal)
<access>	is the set of access letters for that item, as listed under the *ACCESS command.
<User Id>	is the user who submitted the job for printing.
<printer>	is the logical printer selected for the job.
<date1>	is the date that the item was originally created,
<date2>	
<time>	are the date and time that the item was last changed, i.e. written to or saved over in the case of a file, or contents changed in the case of a directory. If the date is today's, then today will appear in the <date1> and <date2> fields.
xx	is the account number associated with the file (see under *ACCOUNT)
yy	is the auxiliary account number associated with the file.
sss	is the station number from which the print job was submitted.
Default=xx/xx	gives the default access letters for a directory (i.e. the access status given to any file saved in it, until this is changed using the *ACCESS command). The default setting may be changed using *DEFACCESS.

The word ...Private only is given to a non-owner of an item, if the access of the item is P.

If this listing is required for all files in a directory, use the \*EX command (see this Section).

## Examples:

\*INFO PROGRAMS

will display a line of information as above for the file (or directory) PROGRAMS.

\*INFO \*

will give information on the first entry (only) in the currently selected directory.

\*INFO \$.FRED

will give information on \$.FRED, which is probably a user root directory for user FRED.

\*INFO

gives information on the currently selected directory.

### **Likely Errors:**

**Not found                      Error D6 (214)**

If the specified item is not found.

### **Associated Keywords:**

\*CATALL, \*EX, \*INFO, SIZER

### **Compatibility Notes:**

Supported by Acorn systems.

---

**Syntax:** **\*LIB** <directory specifier>

### Action with Wild Cards in the Directory Name:

Occurs on first match (alphabetically).

### Description:

This command selects the specified directory as the current library directory.

When any **CHAIN LOAD OPENIN OPENUP \*EXEC \*LOAD \*RUN** or **\*<file specifier>** command is given, the user's currently selected directory will be searched for the file. If it is not found, the *currently selected library* will be searched, and the file opened or loaded from there if found.

The OSFILE machine code call with A=&05 (ROI) and A=&FF (load) (See Econet Advanced User Guide page 38) will also check the library directory if the file specified is not found in the currently selected directory.

Note that the **\*I AM** command (see this Section) will automatically select the library directory selected in the password file for the user on the lowest numbered disc drive, if such a directory exists; hence this command is needed only if a different directory is to be specified as the library. The selected library directory defaults to \$.LIBRARY on logging-on if no particular library was specified in EDITPASS.

See the **\*DISABLE** command in this Section, which makes the library searching equivalent to that in an Acorn File Server (i.e. searched for **\*** and **\*RUN** commands only).

### Examples:

```
*LIB $.OTHERLIB
```

to select this alternative directory as the library.

### Likely Errors:

**xxxx is not a directory**                      **Error 190 (BE)**  
If a file name has been specified

**xxxx Not Found**                              **Error 214 (D6)**  
If the directory specified does not exist.

### Associated Keywords:

**\*ENABLE**

### Compatibility Notes:

Supported on Acorn systems. Note that the library search is done *only* for **\*<file specifier>** (or **\*RUN**) commands, and not for any of the other operations specified above. Acorn systems will automatically select as library a directory called \$.LIBRARY on the same disc as the user root directory is found: this is slightly different from the rules on a SJ Research system, where the directory specified in EDITPASS (usually \$.LIBRARY on the lowest numbered drive) is selected.

---

**Syntax:** `LOAD "<file specifier>"` | `LOAD <string variable>`

where the <string variable> must contain a legal file specifier.

## Action with Wild Cards in the File Name:

Occurs on first match (alphabetically).

## Description:

This BASIC command causes the file, with name equal to the string immediately following the LOAD command, to be copied into memory as though it were a BASIC program. An error message will be generated by the BASIC language system if the file specified did not contain a BASIC program.

The action taken by the File Server is to search through the currently selected directory for the file specified. If it is not found, the File Server will search through the currently selected library (see under \*LIB command), and will load the file from there if found.

## Examples:

`LOAD "COPIER"`

loads this file from the currently selected directory (CSD), or from the currently selected library if it was not found in the CSD. Note that any string constant in BASIC must be enclosed in quotes.

`LOAD FILE$`

loads the file whose name is equal to the string FILE\$. Again the currently selected library is searched if the file is not found in the CSD.

## Likely Errors:

**xxxx is not a file**                      **Error 181 (B5)**  
You cannot LOAD a directory.

**Insufficient access**                      **Error189 (BD)**  
There must be access status **R** for this user, otherwise he will not be able to load the file.

**Bad name**                                  **Error 204 (CC)**  
If the file name contains illegal characters (e.g. \$ % . ^: except in contexts where they are permitted).

**xxxx Not Found**                          **Error 214 (D6)**  
If the specified file did not exist.

## Associated Keywords:

SAVE, \*ENABLE, \*DISABLE

## Compatibility Notes:

Supported by Acorn systems. Note that Acorn File Servers do *not perform a library search with the LOAD command*. The command **\*DISABLE LIBRARY** sets the library search, on an SJ Research File Server, to be the same as the Acorn File Server.

**Syntax:** \*LOAD <file specifier> [<load address>]

## Action with Wild Cards in the File Name:

Occurs on first match (alphabetically).

## Description:

This command causes the contents of the specified file to be copied into memory. \*LOAD is followed by the file specifier, and may optionally take a second parameter, the base address (in hexadecimal) for the copy in memory. If this base address is not specified, the base address will be equal to that in the \*SAVE command (see this Section).

The action taken by the File Server is to search through the currently selected directory for the file specified. If it is not found, the File Server will search through the currently selected library (see under \*LIB command for details), and will load the file from there if found.

## Examples:

\*LOAD DATA

loads this file from the currently selected directory (CSD), or from the currently selected library if it was not found in the CSD. The contents of the file will be loaded at the address recorded by the \*SAVE command when the file was saved.

\*LOAD "FILE\*" FFFF7C00

loads the *first file* (alphabetically) matching the wild card specifier. Again the currently selected library is searched if the file is not found in the CSD. The file will be loaded into the BBC Microcomputer (not any second processor) at address 7C00, which is the base of the Mode 7 screen.

## Likely Errors:

**xxxx is not a file**                      **Error (181) B5**  
You cannot \*LOAD a directory.

**Insufficient access**                      **Error189 (BD)**  
There must be access status R for this user, otherwise he will not be able to load the file.

**Bad name**                                  **Error 204 (CC)**  
If the file name contains illegal characters (e.g. \$ % . ^: except in contexts where they are permitted)

**xxxx Not Found**                          **Error 214 (D6)**  
If the specified file did not exist.

## **Associated Keywords:**

\*SAVE

## **Compatibility Notes:**

Supported by Acorn systems. Note that Acorn File Servers do *not perform* a library search with the \*LOAD command.

---

## Syntax: \*LOGON

### Description:

This program prompts a user for user identifier and password, then logs that user on to a File Server. It is recommended that \*LOGON is used (in a boot file as described below) wherever security is important.

The password is not reflected on the station screen, but an asterisk appears for each keystroke.

This program can be put into file \$.BOOT.!BOOT (and user BOOT should have \*OPT4,3 set), so that any user can press <Shift-break> to run it automatically. See this Section under \*OPT4 for further details. This relies on the File Server being station 254 -- this is the default assumed by the BBC Microcomputer.

\*LOGON is particularly useful with BBC Microcomputers equipped with NFS 3.34, as this does not support the \*I AM <user id.>: option to conceal passwords (See \*I AM command). It also sets the protection byte, so that other users cannot examine the memory of a computer while it is logging on, and erases any copy of the password from the computer memory after log-on.

### Examples:

```
*LOGON
```

```
User: FRED
```

```
Password: *****
```

```
>
```

### Likely Errors:

This program calls \*I AM, and so can cause any of the errors that \*I AM could cause.

### Associated Keywords:

\*I AM, LOGOFF, \*FS, \*BYE, \*PASS

### Compatibility Notes:

Supported by Acorn systems.

---

## Syntax: CHAIN "MULTICOPY"

### Action with Wild Cards in the Directory Name:

Occurs on first match (alphabetically).

### Description:

This program copies entire directory trees between File Servers, or between different places in the same File Server.

It will prompt for the log-on text for the File Server containing the source files, and the same for the destination File Server.

The program will then ask **Do you wish to include sub-directories ?** -- if the user answers **Y**, it will copy the entire set of sub-directories and the files in them. It will also ask if the account information is to be copied -- if the answer to this question is **N**, then all files and directories will be put in the main account of the destination directory.

There is also an option to copy the creation dates of the files; this is intended for use when backing up the File Server. The system manager may set this option so that ordinary users cannot use it.

The user must own the destination directory, and have read access to all the files to be copied.

### Examples:

```
CHAIN "MULTICOPY"  
Multiple file copy utility V1.05
```

MULTICOPY copies groups of files from one File Server to another. It may also be used between directories or discs on the same File Server.

```
Log-on text for source FS:  
*I AM 254 FRED
```

```
Log-on text for dest. FS  
(or press RETURN for same FS):  
*I AM 253 FRED  
Do you wish to include sub-directories (Y/N): Y  
Do you wish to copy account information (Y/N): Y  
Do you wish to copy creation date etc.  
(for system manager's use only) (Y/N): N  
source directory name      : PROGS  
destination directory Name: PROGS
```

.  
.  
.

(list of the files being copied)

The next example shows a copy of the directory structure '\$.RELEASE' from a hard disc called 'MAIN1' to a floppy disc called 'Main2'.

## CHAIN "MULTICOPY"

Multiple file copy utility V1.05

MULTICOPY copies groups of files from one File Server to another. It may also be used between directories or discs on the same File Server.

Log-on text for source FS:

\*I AM 254 FRED

Log-on text for dest. FS

(or press RETURN for same FS):

\*

Do you wish to include sub-directories (Y/N): **Y**

Do you wish to copy account information (Y/N): **N**

Do you wish to copy creation date etc.

(for system manager's use only) (Y/N): **N**

source directory name : **\$\*1.RELEASE**

destination directory Name: **\$\*2.NEWREL.ANOTHER**

.  
.  
.

(list of the files being copied)

## Likely Errors:

### Insufficient access

#### Error189 (BD)

If the user does not have access R to all the files to be copied from the source, or does not own the destination directory.

### xxxx is not a directory

#### Error 190 (BE)

If the user has specified a file as the source or destination directory name prompt.

### Already opened by xxxx

#### Error 195 (C2)

MULTICOPY will save over a file of the same name. If this file was already open, this error will occur.

### Locked

#### Error 195 (C3) Locked

After an attempt to save over a file of the same name, if the latter was locked.

### xxxx Not Found

#### Error 214 (D6)

If the source or destination directories could not be found.

### Account xxxx bankrupt

#### Error 198 (C6)

If the account number being saved to does not have sufficient credit.

## Associated Keywords:

COPIER

## Compatibility Notes:

Supported by Acorn systems, except that accounts do not exist, and so an attempt to copy account information across will cause an error. Since Acorn systems use the root of user's tree of directories to determine its ownership (rather than account numbers), a user will not have owner access to files specified as \$.<directory name>.<file name>. To get round this problem, it is wise to log on as a system privileged user.

---

**Syntax:** \* NOTIFY <station number> <text> | \*NOTIFY <User Id> <text>

## Action with Wild Cards in User Identifier:

Occurs on first match, from the top of the list as produced by \*USERS (see this Section) downward.

## Description:

This utility produces the message \*| <sender station number>: <text> \_\_ on the screen of the specified station, or of the station where a specified user is logged-on.

If the specified user is logged-on at several stations, the message will appear only at one of them: this will be the station at which the specified user last performed a filing operation, and also the first one that would appear in the list produced by the program \*USERS (see this Section).

The station specified (or found by the above process from the user identifier) must be switched on and connected to the network, otherwise the error message **Not listening** will be given. If the user has run \*PROT or otherwise set the protection byte in his computer, the **Not listening** message will also be produced.

## Examples:

\*NOTIFY FRED HELLO THERE

will send the message HELLO THERE to the station at which user FRED last performed a network filing operation.

\*NOTIFY 3 TIME FOR LUNCH

will send the message to station 3, if it is switched on. The message produced in each case will be of the form (if the message had been sent from station 5)

\*|| 005: TIME FOR LUNCH

The message will be accompanied by a <ctrl-G> character, which produces a beep.

## Likely Errors:

### Net Error

### Error 161 (A1)

If the line is more than 80 characters in length. This is due to overflow of the File Server command line buffer. Use several \*NOTIFY commands for long messages, or use a program to call OSWORD with A=&14 (see Chapter 8).

### Not listening

### Error 162 (A2)

If the destination station is switched off, or if its protection byte has been set.

### Not logged on

### Error 174 (AE)

If a user identifier was specified, and that user was not logged on at the currently selected File Server.

## Associated keywords:

\*PROT, FORCER

## Compatibility Notes:

Supported on Acom systems, but note that the user table is not re-ordered to reflect the latest operation, but is effectively in random order, so that \*NOTIFY <User Id> will not necessarily find that user.

---

**Syntax:** <numeric variable>=OPENIN "<file specifier>"  
<numeric variable>=OPENIN <string variable>

### Action with Wild Cards in the File Name:

Occurs on first match (alphabetically).

### Description:

This BASIC command opens a file for random access using BGET#, INPUT# commands. OPENIN is followed by a file specifier, and is an integer function, returning a value called the *channel* for the file: *if the channel is zero, this means that the file was not found, and no other error will be given.* OPENIN opens a file for reading only: an attempt to write to the file will cause an error. The user must have access **R** to the file to be able to open it for input.

The action taken by the File Server is to search through the currently selected directory for the file specified. If it is not found, the File Server will search through the currently selected library (see under \*LIB command for details), and will open the file from there if found. If a library search is *not* wanted, then the programmer should specify either the full file specifier (beginning with \$), or use @ to specify the currently selected directory. See also \*DISABLE LIBRARY command in this Section, to restrict the library search.

A file may be opened for input even if other users have already opened it for input also. However, it is not permitted to open a file for input if it already open for output or update.

It is also permissible to open a *directory* using this command, but it is not possible to read any data from it: an attempt to do so will cause an error. This keyword could be useful to check for the existence of a directory. Note that OPENIN will *not* search the library for a directory (but it will, as normal, for a file).

There is a limit to the number of files that a user may have open at once. This is initially 8, but the user root directory (URD) and currently selected library are opened by the system at log-on, leaving a maximum of 6 for the user. In addition, if the user specifies a currently selected directory other than the URD or library, this is also opened; also the \*DIR command uses a further channel fleetingly. Hence it is wise for programmers to rely only on having four channels for random access filing operations.

*The OPENIN command in BASIC 1 is in fact OPENUP.* This means that you cannot use it to open a file to which you have access **R** only, if your BBC Microcomputer is equipped with BASIC 1. To find which version is fitted, type \*BASIC followed immediately by **REPORT**. The response will be either **1981** for BASIC 1, or **1982** for BASIC 2. To open a file for reading only, use the function FNOpenin given below.

Note also that the command OPENIN in BASIC 1 has the same internal representation as the command OPENUP in BASIC 2. This means that a file initially run and saved in BASIC 1 will run identically on BASIC 2 -- furthermore, when the program is listed in BASIC 2, the translation to OPENUP will have happened automatically.

However, a BASIC 2 program containing OPENIN commands will *not run in BASIC 1*, and these commands will not appear in a BASIC 1 listing -- hence if compatibility is required between BASIC 1 and BASIC 2 for opening read-only files, the function FNOpenin should always be used.

### Examples:

A%=OPENIN "DATAFILE"

followed later in the program by:

INPUT#A%, X

reads the value of X from the file DATAFILE. The system will look in the currently selected library for the file, and open that, if the file is not present in the currently selected directory.

```
channel%=OPENIN"$ .JOHN.PROJECT.DATA"
```

sets up to read from a file in sub-directory PROJECT of directory \$.JOHN (and will not search the library if file DATA is not found). Similarly,

```
channel%=OPENIN"@.data23"
```

will look for **data23** only in the currently selected directory, and not in the library.

### Program FNOpenin for use with BASIC 1:

The BASIC 1 language does not support OPENIN as described here. It contains a keyword OPENIN, but it is in fact OPENUP (see this Section). The function FNOpenin listed below is recommended if you need to open a file for reading only (in particular, when the user has only access **R** to the file).

At the head of the program:

```
DIM combuf 128  
osfind=&FFCE
```

and after the body of the program:

```
DEF FNOpenin($combuf)  
LOCAL A%, X%, Y%  
A%=&40  
X%=combuf  
Y%=X% DIV 256  
=(USR osfind) AND &FF
```

Then, in the body of the program, use (for example):

```
channel%=FNOpenin("Data")
```

then check that channel% is non-zero, and

```
byte=BGET#channel%
```

### Likely Errors:

#### Insufficient access                      Error 189 (BD)

There must be access status **R** for this user, otherwise he will not be able to open the file for input. If the BBC Microcomputer is equipped with BASIC 1, it will be necessary to use the program given above.

#### Too many files open                      Error 192 (C0)

There is a limit to the number of channels available, normally 5 or 6.

#### File not open for update                Error 193 (C1)

Will be caused by an attempt to use BPUT# or PRINT# after OPENIN.

#### Already opened by xxxx                  Error 194 (C2)

It is not permitted to open a file for input, if it already open for output by either this user or another.

#### Bad name                                  Error 204 (CC)

If the file name contains illegal characters (e.g. \$ % . ^: except in contexts where they are permitted).

### Associated keywords:

OPENUP, OPENOUT, \*CLOSE, BGET, BPUT

## Compatibility Notes:

Supported on Acorn systems. Note that Acorn File Servers do **not** perform a library search with the OPENIN command.

**Syntax:** <numeric variable>=OPENUP "<file specifier>" |  
 <numeric variable>=OPENUP <string variable>

### Action with Wild Cards in the File Name:

Occurs on first match (alphabetically).

### Description:

This BASIC command opens a file for random access using BGET#, BPUT#, INPUT#, PRINT# commands. OPENUP is followed by a file specifier, and is an integer function, returning a value called the *channel* for the file: if the channel is zero, this means that the file was not found, and no other error will be given. OPENUP opens the file for reading or writing.

The user must have at least access **W** to the file to use this command, and would normally have access **R** as well, so that he could read from or write to the file. There is however a meaning to write-only access: this is *append only* -- the file may only be written to if the pointer PTR# is equal to the extent of the file EXT#.

The action taken by the File Server is to search through the currently selected directory for the file specified. If it is not found, the File Server will search through the currently selected library (see under \*LIB command for details), and will open the file there if found. To inhibit the library search, either specify the full file name (beginning with \$), or use @.<file name>. See also \*DISABLE LIBRARY command in this Section, to restrict the library search.

A file may be not opened for update if it has already been opened for input or update by this or another user.

There is a limit to the number of files that a user may have open at once. This is initially 8, but the user root directory (URD) and currently selected library are opened by the system at log-on, leaving a maximum of 6 for the user. In addition, if the user specifies a currently selected directory other than the URD or library, this is also opened; also the \*DIR command uses a further channel fleetingly. Hence it is wise for programmers to rely only on having four channels for random access filing operations.

This keyword does not exist in BASIC 1, but the BASIC 1 keyword OPENIN has exactly the same effect. To find which version is fitted, type \*BASIC, followed immediately by REPORT. The response will be either 1981 for BASIC 1, or 1982 for BASIC 2.

Note also that a program run in BASIC 2, containing OPENUP commands, will run satisfactorily on BASIC 1 -- when it is listed in BASIC 1 the commands will have changed into the BASIC 1 keyword OPENIN (see this Section).

### Examples:

```
A%=OPENUP "DATAFILE"
```

followed later in the program by a check that A% is non-zero, then:

```
PRINT#A%, X
```

writes the value of X to the file DATAFILE. The system will look in the currently selected library for the file, and open that, if the file is not present in the currently selected directory.

```
channel%=OPENUP "$ .JOHN .PROJECT .DATA"
```

sets up to read from a file in sub-directory PROJECT of directory \$.JOHN

## Likely Errors:

**xxxx is not a directory                      Error 181 (B5)**

You cannot OPENUP a directory.

**Insufficient access                      Error 189 (BD)**

There must be access status **W** for this user, otherwise he will not be able to open the file for for updating.

**Too many files open                      Error 192 (C0)**

There is a limit to the number of channels available, normally 5 or 6.

**File not open for update                  Error 193 (C1)**

Caused after OPENUP by a user only having access **W** to a file attempting to read the file, or to write to the file without PTR# being equal to EXT#.

**Already opened by xxxx                  Error 194 (C2)**

It is not permitted to open a file for update, if it already open for any purpose by either this user or another.

**Bad name                                  Error 204 (CC)**

If the file name contains illegal characters (e.g. \$ % . ^: except in contexts where they are permitted)

## Associated keywords:

OPENUP, OPENOUT, \*CLOSE, BPUT, BGET

## Compatibility Notes:

Supported on Acorn systems. Note that Acorn File Servers do *not* perform a library search with the OPENUP command, nor do they support the 'append only' feature resulting from **W** only access to a file.

---

**Syntax:** <numeric variable>=OPENOUT "<file specifier>" |  
 <numeric variable>=OPENOUT <string variable>

### Action with Wild Cards in the File Name:

Wild cards prohibited.

### Description:

This BASIC command creates a new file with the name <file specifier>. It will delete any existing file of the same name. The file is opened for reading or writing, although it will be necessary to write some data first, otherwise an EOF error will occur. It will be possible to write to the file using BPUT# or PRINT# even if the default access for the directory containing the file is **R** only.

OPENOUT is an function, returning a value between 0 and 255. If this number is non-zero, it is called the *channel* for the file. A value of zero means that a directory specified was not found -- zero is not returned for any other reason (and can therefore be used to check for the presence of a directory).

The length initially assumed for the file is zero: file space will be allocated according to the value of PTR#. The minimum non-zero file size is 1 kilobyte, and the length will be increased in kilobyte steps when PTR# becomes more than the file size. Note that the File Server will allocate 1 Kbyte pages *only* when they are required (i.e. when they are written to). Hence it is possible to have a file with extent of 10 Kbytes, but only using 2 Kbytes of disc space, because there is data only near the beginning and end.

There is a limit to the number of files that a user may have open at once. This is initially 8, but the user root directory (URD) and currently selected library are opened by the system at log-on, leaving a maximum of 6 for the user. In addition, if the user specifies a currently selected directory other than the URD or library, this is also opened; also the \*DIR command uses a further channel fleetingly. Hence it is wise for programmers to rely only on having four channels for random access filing operations.

### Examples:

A%=OPENOUT "NEWDATA"

followed later in the program by:

BPUT#A%, CH

which will write the least significant single byte of CH to the file NEWDATA.

### Likely Errors:

**xxxx is not a directory**                      **Error 181 (B5)**

Caused by attempting to create a file with the same name as that of a directory.

**Insufficient access**                      **Error 189 (BD)**

Caused if the user is not an owner (i.e. does not have access to the main or auxiliary account) of the directory in which the new file is to be created.

**Too many files open**                      **Error 192 (C0)**

There is a limit to the number of channels available, normally 5 or 6.

**Already opened by xxxx**                      **Error 194 (C2)**

If this or another user has this file open for reading or writing, then it cannot be deleted by creating a file with the same name, until it has been closed.

**Entry locked**                      **Error 195 (C3)**

It will be necessary to use the \*ACCESS command to unlock the file, before it can be deleted by creating a

file with the same name.

**Bad name**

**Error 204 (CC)**

If the file name contains illegal characters (e.g. \$ % . ^: except in contexts where they are permitted).

**Associated keywords:**

OPENI, OPENOUT, \*CLOSE, BPUT, BGET

**Compatibility Notes:**

Supported by Acorn systems, but the space allocation works differently.

**Syntax:** \*OPT1, <number>

## Description:

This command controls the display of information after any filing operation. The value of the number can be 0 or 1, and has the following effect:

*OPT1, 0	information displayed.	No
*OPT1, 1	following information is displayed after any <b>LOAD SAVE *LOAD *SAVE *RUN</b> or <b>*&lt;file specifier&gt;</b> commands only. <file name> <load address> <execute address> <length>	The

The addresses and length are as displayed in \*INFO or \*EX, and are in hexadecimal. Specifying a <number> in excess of 1 has the same effect as \*OPT1,1. (Note that this is true only of the Econet system, and not necessarily of other filing systems for the BBC Microcomputer).

The setting of OPT1 remains only for the duration of the session, although it is preserved over <Break>. The setting is lost if the power is turned off to the BBC Microcomputer, or if <Ctrl-Break> is pressed.

## Examples:

\*OPT1, 1

sets the flag to cause additional information to be printed after certain filing operations (as above).

## Known Bug:

In NFS 3.6, if OPT1,1 is set, and printing through the network is in progress (i.e. character <Ctrl-B> has been output), then it is possible that a LOAD or SAVE operation that fails will *not* generate an error message to the user (in fact this occurs when the network printer buffer is nearly full). Care should be taken with the use of OPT1,1 if printing is likely.

## Likely Errors:

**Bad command**                      **Error 254 (FE)**  
If the second number is in excess of 255.

## Associated keywords:

LOAD, \*LOAD, SAVE, \*SAVE

## Compatibility Notes:

Supported on Acorn systems. Some versions will not work with the value of <number> greater than 1.

---

**Syntax:** \*OPT4, <number>

## Description:

This command controls the automatic execution of commands when a user logs on using the \*I AM command. The number after \*OPT4 can be between 0 and 3, and has the following effect:

- \*OPT4, 0 specifies no action at log-on
- \*OPT4, 1 performs \*LOAD !BOOT i.e. the file is loaded into memory, but no further action is taken
- \*OPT4, 2 performs \*RUN !BOOT i.e. runs !BOOT as though it were a machine code program
- \*OPT4, 3 performs \*EXEC !BOOT, i.e. it obeys commands in !BOOT as though they had been typed from the keyboard. This is probably the most useful option, as it allows File Server and other commands to be put easily into the boot file.

If \*OPT4,x has been set at some time with x not zero, the File Server will search for the file !BOOT at log-on. The User Root Directory will be searched first, and then the Library directory. If !BOOT is not found, the message **File not found** will be displayed: note that this does *not* mean that log-on has failed, but only that !BOOT was absent. (Note also that SJ Research File Servers do not usually give **File not found** -- under most other circumstances <file name> not found will appear).

If the only file !BOOT in the system is in the library, then it would be possible to use it to produce a message of the day at log-on, which would be displayed every time a user (who has \*OPT4,3 set) logs on.

The system manager can, for each user, lock the boot option so that the user cannot change it -- an attempt to do so will cause error BA (see below).

## Examples:

\*OPT4, 2

sets the auto-run option so that file !BOOT is loaded and run as a machine code program at log-on.

## Likely Errors:

### Insufficient privilege                      Error 186 (BA)

There is an option available for the system manager to 'lock' the password and \*OPT4 setting, so that a user cannot change them. If this option has been set, this error will be given.

### Password file changed                      Error 3 (03)

Caused if the system manager has made any change to the password file since the user logged on. The user should log on again.

## Associated keywords:

\*I AM, \*BUILD !BOOT

## Compatibility Notes:

Supported on Acorn systems, but note that the library search is performed *only* after \*OPT4,2 is set with Acorn File Servers.

---

**Syntax:** OSCLI "<string>" | OSCLI <string variable>

### Description:

This command is not associated specifically with the Econet system, but this description is supplied because it does not appear in any other official documentation.

OSCLI is implemented only on BBC BASIC 2, and not on BASIC 1. Its effect is to pass the string specifier direct to the operating system Command Line Interpreter (CLI), so that OSCLI "<string>" has the same effect as \*<string>.

Unlike \* commands in BASIC, OSCLI can be used within a multi-statement line, and with string variables. This means that programs can accept \* commands in place of their normal input, and then execute them.

For programs that need to run on BASIC 1, the procedure PROCoscli defined below should be used.

### Procedure for use with BASIC 1:

```
DIM buf% 128
```

within the main body of the program, followed by, after the end,

```
DEF PROCoscli ($buf%)  
LOCAL X%, Y%  
X%=buf%  
Y%=X% DIV 256  
CALL &FFF7  
ENDPROC
```

The procedure could be called by (for example):

```
270 INPUT source$  
280 IF LEFT$(source$)="*" THEN PROCoscli(source$): GOTO 270
```

### Likely Errors:

There are no errors specific to this command, but any error generated by the command sent to the CLI will occur as normal.

### Associated keywords:

\*I AM, EDITPASS

### Compatibility Notes:

Supported on Acorn systems.

---

**Syntax:** \*PASS <old password> <new password>

## Action with Wild Cards in the Password:

Wild cards prohibited.

## Description:

This command changes the user's password. The system manager may have set a password up for you, in which case you must quote the old password followed by the new one. If no password had been set then it is necessary to quote a null string "" as the old password.

Passwords have a maximum length of 10 characters, and may contain all characters permitted in file names, i.e. alphanumeric and ! - \_ . Upper and lower case letters are treated as equivalent.

The system manager can set an option to prevent you from changing your password. If this has been done, you will get insufficient privilege (error BA) if you attempt to change it.

If security is important to you, then take care in your choice of a password. We recommend against using any character string that others might guess (e.g. your wife's name, your phone number, car number, etc.). It is probably best not to use a normal word, since someone may see part of it and then guess the rest. Finally, use as many characters as possible (it does not take long to run a program to try all 4-character passwords).

## Examples:

To set up your password as HELLO type:

```
*PASS "" HELLO
```

(assuming no password set previously). Conversely, to clear the password, type:

```
*PASS HELLO ""
```

## Likely Errors:

**Password file changed**                      **Error 3 (03)**

This error will be produced if the password file has been changed by the system manager, while the user is logged on. The user should log on again.

**Bad password**                                **Error 185 (BD)**

There is an illegal character in the password quoted, probably \* # \$ % ^:

**Wrong password**                            **Error 187 (BB)**

The old password does not match the one stored.

## Associated keywords:

## Compatibility Notes:

Supported by Acorn systems. Note that the option to 'lock' the password does not exist on Acorn File Servers.

**Syntax:** **\*PATHNAME**

## **Description:**

This program prints the full path of the currently selected directory. Thus you can check where you are in a directory structure and which disc you are using.

**Examples:**  
lh\*PATHNAME  
:MAIN-IV.\$FS-manual.Iss023

The currently selected directory is on the disc MAIN-IV and is the directory Iss-023 in the directory FS-manual in the disc root directory.

## **Likely Errors:**

There are no errors specific to this command.

## **Associated keywords:**

## **Compatibility Notes:**

Supported by Acorn systems.

## Syntax: \*PRINT

### Description:

The Econet software in the BBC Microcomputer assumes that the network printer server has number 235 unless specified otherwise by this command. The SJ Research File Server contains a printer server, so if the File Server is station 254 and the built-in printer server is used, the \*PRINT (or \*PS command below) command will be necessary to get any output. See Sections 5.5 and 6.5 for a complete explanation of printing through the network.

\*PRINT is now superseded by the command \*PS below, which also performs automatic selection of printer servers if there is more than one in a network, and allows the user to specify printers by name. \*PRINT will only work if the File Server you are logged on to is the printer server you wish to use.

\*PRINT combines the effects of \*FX 5,4 (select network printer), \*FX 6,0 (allow line-feeds through), and \*PS <File Server station number> described above. It thus performs all the commands usually required to select the network printer.

Users who have automatic line feed selected on their printers can modify this program to remove the \*FX6,0. To do so, log on as a system privileged user, then type:

```
*DIR $.LIBRARY
H%=OPENUP "PRINT"
PTR#H% = &A
BPUT#H%,10
CLOSE#H%
```

This modification may have already been made, if it is needed for the printers on your File Server.

### Likely Errors:

There are no File Server errors specific to this program.

It is possible that any program calling \*FX5 may not work if the local printer buffer is not empty -- if, for example, a local printer has been used but had not completed the job. If \*PRINT appears to hang up, press <Escape> <Ctrl-C> <Escape>, then try \*PRINT again.

### Associated keywords:

\*PS

### Compatibility Notes:

Supported by Acorn systems. Note that \*PRINT only works where the File Server is also the printer server -- but it is recommended that they are combined in Level 2 systems.

---

**Syntax:** \*PRINTER || \*PRINTER <logical printer name>

## Action with Wild Cards in the Printer Name:

Wild cards prohibited.

## Description:

This command is intended for use with the \*PRINT and \*PRINTOUT commands (see this Section). The user must be logged on to a File Server, and the \*PRINTER command will apply to this File Server.

Used without an argument, this command displays the name and status of the currently selected logical printer. The possible statuses are:

ready	the printer is available for use.
busy	the printer will be available for use when the current work is finished.
jammed	the printer is jammed or has run out of paper, ribbon etc. This message is also given if the print queue directory is full.
not authorised	this user is not authorised to use the selected printer.

If \*PRINTER is followed by a printer name, then the internal logical printer selection, within the current File Server for the user's station, will be changed to the new name. Error 0A will be produced if the user is not authorised to select the printer specified. See Section 2.5.5 for a description of logical printers.

\*PRINTER will *not* change the printer server station number in the BBC Microcomputer, but if the printer server station number has been selected to be the same as the File Server, it will change the File Server's internal selection, and therefore will change which printer is used for network printing.

## Examples:

\*PRINTER

will produce, for example:

EPSON ready

Similarly the command:

\*PRINTER ML

will produce a response of the form:

ML busy with station 003 (FRED)

## Likely Errors:

**Not authorised to use printer Error 10 (0A)**

If the system manager has restricted the use of this printer to users with access to a certain account.

## Associated keywords:

\*PS, \*PRINTOUT

## **Compatibility Notes:**

Not supported by Acorn systems.

**Syntax:** \*PRINTOUT <file specifier>

## Action with Wild Cards in File Name:

Occurs on first match only (alphabetically).

## Description:

This command causes the specified file to be printed, at the currently selected logical printer as selected by \*PS (i.e. the same printer that would be used if normal printing were selected), on the current File Server. The appropriate printer server banner and end-text characters (as set up for the logical printer by the system manager) will be printed.

\*PRINTOUT therefore allows a program that generates printed output to run even if the printer is currently busy. Instead of sending output directly to the printer (by \*FX5,4 or \*PS followed by <Ctrl-B>), the user can spool the screen output to a file by typing \*SPOOL <file specifier>, and terminating the file by typing \*SPOOL on its own. When the printer becomes free, the user should then type \*PRINTOUT <file specifier>, which will send it the contents of the file. It is recommended that users run \*PUTGET before running \*SPOOL -- see notes under \*SPOOL in this Section.

Another advantage of \*PRINTOUT over standard printing is that the user is informed if there is any problem. Sending <Ctrl-B> (to start printing through the network) while the printer is busy, will cause the user's BBC Microcomputer to hang for about 30 seconds, before coming back with **Not listening**. The error messages generated by \*PRINTOUT are more informative as well as appearing immediately.

There is a possible hazard with \*PRINTOUT, if your printer does an automatic line-feed whenever it receives a carriage return character (ASCII value &0D). When doing normal printing from a BBC Microcomputer, line-feed characters (ASCII &0A) will be filtered out by the BBC Microcomputer (unless this is changed with the \*FX6 command, as described in the BBC User Guide, page 408). \*PRINTOUT does not do this filtering, which means that text containing line feed characters will be double spaced on a printer which does auto-line-feed.

The solution is to turn off the auto-line-feed option on the printer, and type \*FX6,0 at each BBC Microcomputer (this can be done in a boot file whenever a user logs on). The behaviour of \*PRINTOUT will then be consistent with normal printing.

Note also that the text files produced by many word processors do not contain line-feed characters. If \*PRINTOUT is to be used with such text files, the user should start a spool file (as above), then print the text to the computer screen, then use \*PRINTOUT with the spool file.

## Examples:

```
*PRINTOUT TEXTFILE
```

will send the file TEXTFILE to the printer.

```
*SPOOL LISTING  
LIST
```

```
  .  
  .  
(BASIC listing on screen)
```

```
  .  
  .
```

\*SPOOL  
\*PRINTOUT LISTING

will send a listing of the BASIC program in memory, to the printer.

## Likely Error

**Printer busy with station nnn(xxxx) Error 9 (09)**

If the printer is not currently available.

**Not authorised to use printer Error 10 (0A)**

The system manager can set up a printer, so that only users with access to a certain account number can use it.

**Insufficient access Error 189 (BD)**

The user must have read access to the specified file.

## Associated keywords:

\*FLUSH, \*PRINTER, \*REROUTE

## Compatibility Notes:

Not supported by Acorn systems.

---

## **Syntax: \*PROT**

### **Description:**

This program protects the station at which it is entered against low level operations such as PEEK and POKE from other stations, and also network commands like \*REMOTE and \*NOTIFY. Other machines attempting these operations will behave as if the PROTECTED station were not there. Protection continues until <Ctrl-Break> is pressed or the command \*UNPROT is given. More information about this call is given in the machine code section, Chapter 10.

If some operations from other stations are required but not others, an extended version of this command is given by \*PROTEX.

### **Likely Errors:**

There are no errors specific to this command.

### **Associated keywords:**

\*PROTEX, \*UNPROT

### **Compatibility Notes:**

Supported by Acorn systems.

---

## **Syntax: \*PROTEX**

### **Description:**

This program offers the user the option of protecting his station against various types of operation from other stations in the network. The user will be prompted with text of the form:

Protection against: Halt (Y/N)

The options of protection against Halt, Utils, Prc, Jsr, Poke and Peek are offered. Responses other than Y or N will be ignored, except <Escape> which causes the machine to hang up, and <Break> which leaves the \*PROTEX program.

The Halt option stops all action in the machine until a Continue command is issued and is thus a good thing to be protected against.

Utils protects against fileserver utilities which affect other stations, in particular FORCER, \*NOTIFY, \*VIEW and \*REMOTE.

Proc disallows all remote procedure calls to a station. Only a few programs use these calls (one is \*FAST) and they are not usually implemented without the station user's co-operation, so protection against these is not often needed.

Jsr prevents other stations from taking over areas of your station's memory to run a subroutine for one of their programs. This subroutine will need to have been entered into your station by using POKE.

The Poke and Peek options respectively prevent other stations from writing to or reading from your station's memory directly.

The \*PROT command has the effect of setting all of these protections on; whereas \*UNPROT sets them all off.

### **Likely Errors:**

There are no errors specific to this command.

### **Associated keywords:**

\*PROT, \*UNPROT

### **Compatibility Notes:**

Supported by Acorn systems

---

**Syntax:** **\*PS** <station number> | **\*PS** | **\*PS** <logical printer name>

### Action with Wild Cards in the Printer Name:

Wild cards prohibited.

### Description:

The Econet software in the BBC Microcomputer assumes that the network printer server has number 235 unless specified otherwise by this command. The SJ Research File Server contains a printer server, so if the File Server is station 254 and the built-in printer server is used, the **\*PS**, or **\*PRINT**, commands will be necessary to get any output. This program will also perform the call **\*FX5,4** to select the network printer. The program can also have the effect of a **\*FX6,0** call if the printer in your network is not set to do automatic line feeds. (The system manager will configure the program to suit the printer.)

If there are several stations with printer servers on the network, the **\*PS** program can be used to search for an unoccupied printer server. Typing **\*PS** on its own will select the first free printer available, or typing **\*PS** <logical printer name> will search for a printer server with the specified logical printer. A full discussion of printing through the network is given in Sections 2.5 and section 6.

The system manager will allocate specific names to particular printer/banner combinations, and the use of **\*PS** <logical printer name> will search for a combination with that particular name. There can be several logical printers with the same name, but these should be identical. With current versions of software you must be logged on to a File Server to be able to use **\*PS** <printer name>.

After **\*PS** (either with a name or with no argument), the system will display the station numbers of all the printer servers that responded, followed by **Station nnn selected**, or **No station responding** if a corresponding printer cannot be found.

Note that **\*PS** <printer name> or **\*PS** (without an argument) will work only if the printer servers (in BBC Microcomputers) are equipped with NFS 3.6 or later software. With NFS 3.34 in the printer server, the search will not work, but the use of **\*PS** <station number> will function normally.

### Examples:

There may be a daisy-wheel printer with name **DW**, and some Epson dot-matrix printers with name **EPSON**. To find a free dot-matrix printer, the user could type

```
*PS EPSON
```

The system would then reply either (for example):

```
Station 235 busy
Station 234 ready
Station 234 selected
```

or

```
No station responding
```

### Likely Errors:

There are no errors specific to this program. If the printer server station is specified explicitly, no check is

made on its existence until printing is attempted, when Not listening Error 162 (A2) or No reply Error 165 (A5) will occur if the printer server is either busy or non-existent.

It is possible that any program calling \*FX5 may not work if the local printer buffer is not empty -- if, for example, a local printer has been used but had not completed the job. If \*PS appears to hang up, press <Escape> <Ctrl-C> <Escape>, then try \*PS again.

### **Associated keywords:**

\*PRINTER, \*PSLIST

### **Compatibility Notes:**

Supported by Acorn systems, but only the \*PS <station number> or \*PS (no argument) versions. Note also that Acorn versions of the \*PS command may work differently.

---

## Syntax:\*PSLIST

### Description:

\*PSLIST displays a list of all of the eligible printer servers on the network, i.e. those you are currently allowed to use. For SJ Research printer servers, the logical printers available on each File Server will be listed after the station number.

Logical printers will not be listed if they have been set by the system manager to be non-existent. However, all other logical printers on an eligible printer server will be listed, even if you are not allowed to use them.

If the Eiconet installation comprises multiple networks, \*PSLIST will also display printer servers on other networks, preceded by their network number (e.g. printer server 235 on network 2 will be displayed as 002.235).

The status of the physical printer appropriate to the currently selected logical printer will be given by \*PSLIST after the printer server station number. The status codes currently supported are:

**Ready** -- this printer is ready to start printing, or that print spooling is available for this printer.

**Busy with nnn** -- this non-spooling printer is busy printing output from station nnn.

**Jammed** -- this printer has jammed with paper, has run out of ribbon or some similar event. For a print-spooling printer, the directory %PRINTQ may be full or not found. Jammed printers will not accept any data.

If you are not allowed to use the logical printer which is currently selected for you on a printer server, \*PSLIST will not list that station. Note that it is not possible to select a logical printer you cannot use with the commands \*PS and \*PRINTER; so there are only three reasons why a printer server should not be listed:

1. You may not be allowed to use any logical printers on that printer server. The logical printers may not exist, or they may require users to be logged on the File Server or to have access to a particular account.
2. You may not have changed your logical printer on that printer server since you logged on, and the default printer on that File Server may not be available to you.
3. You may have selected a logical printer on that printer server with \*PS or \*PRINTER when you had access to it; but the system manager has edited the printer information so that you are no longer allowed to use that logical printer.

### Examples:

```
*PSLIST
```

```
235 Ready
simple
fancy
nobann
200 Busy with 023
```

### **Likely Errors:**

There are no errors specific to this command.

### **Associated keywords:**

\*PRINTER, \*PS

### **Compatibility Notes:**

Supported by Acorn systems, except that Acorn systems do not support logical printers.

---

**Syntax:**\*PTIME | \*PDATE | \*PDATE2

## **Description:**

These programs print the time of day or the date, without preceding or following new line characters. They are intended for use from a BASIC or other program, and allow the user to enclose the output in text if required, as shown in the example below.

The printed format from PTIME is **hh:mm**

The printed format from PDATE is **dd/mm/yy**

The printed format from PDATE2 is the date in text form, for example **16th February, 1984**

## **Examples:**

```
340 PRINT "The time is now";
```

```
350 *PTIME
```

```
360 PRINT "hours past midnight"
```

Note that, whilst lines 340 and 350 may be combined with a colon, the following text must be on its own line if PTIME is called in this way. This is because BASIC (or any other language system) passes the entire remainder of the line to the operating system when it meets a "\*" character. (In BASIC 2, you could use OSCLI("PTIME") in place of \*PTIME, to avoid this problem).

## **Likely Errors:**

There are no errors specific to this program. A nonsensical time output implies that the real-time clock has not been set, or that the File Server has been switched off for several months, so that the clock battery has run down. Ask your system manager for help.

## **Associated keywords:**

\*GTIME, SETTIME

## **Compatibility Notes:**

Supported on Acorn systems. Level 2 File Servers have only the date; level 3 File Servers have both date and time, and they use the interval timer function in the BBC Microcomputer thereafter -- there exists the possibility that this timer could lose time under certain circumstances.

---

## Syntax: \*PUSER

### Description

This program prints the name of the user logged on to the station on the screen. This is not followed by a carriage return so more text can be added directly after it. \*PUSER can be used in cases of amnesia or to personalise printed output from programs.

User names have a maximum length of 10 characters.

### Examples:

```
10PRINT "Hello ";
20 *PUSER
30 PRINT ", you are wonderful !"
```

This is similar to the command \*GUSER which allows the user name to be read into a string.

Note that, whilst lines 10 and 20 may be combined with a colon, the following text must be on its own line if \*PUSER is called in this way. This is because BASIC (or any other language system) passes the entire remainder of the line to the operating system when it meets a "\*" character. (In BASIC 2, you could use OSCLI("PUSER"), to avoid this problem).

### Likely Errors:

There are no errors specific to this command.

### Associated Keywords:

\*CV, \*FS, \*GUSER, \*I AM

### Compatibility Notes

Supported by Acorn systems, but note that some Acorn File Servers support user names of more than 10 characters.

# \* PUTGET AND \* PUTGET2

Utility program  
Must be run before a loading user program.

**Syntax:** \*PUTGET |  
          \*PUTGET2

## Description

PUTGET is a program which resides in memory (it reserves some private workspace for itself rather like a sideways ROM) and implements *local buffering* of random access operations to files. That is it 'packages up' BGET and BPUT calls into blocks of 64 bytes, then uses the OSGBPB filing system call. This results in a speed increase by a factor of roughly 64; as it takes almost as much time to read or write a single byte as to read a block of 64 bytes, due to network and File Server overheads.

To activate simply type **\*PUTGET**, and a banner will be printed. PUTGET will remain active (even over BREAK) until you specifically de-activate it. The most noticeable effect of PUTGET is to move the BASIC value PAGE up by three pages, and because of this *it will corrupt any BASIC program that is in RAM*. Users should therefore be sure to run PUTGET *before* loading any BASIC programs.

The use of PUTGET will *significantly speed up programs* that use the BASIC commands BPUT# BGET# INPUT# PRINT#, and the OS commands \*EXEC \*SPOOL and the DFS Utilities (if fitted) \*TYPE \*DUMP \*LIST \*BUILD.

Running PUTGET twice will produce the message **PUTGET already active** -- this message is just printed and does not cause an error.

A special version **PUTGET2** is provided for use with word processors such as WORDWISE. This version loads into the RS423/cassette tape filing system buffer, and provides more space for text. Note however that this is illegal workspace, and PUTGET2 corrupts these buffers. PUTGET2 should therefore not be used except with Wordwise etc., and especially not if it is intended to use the serial input or output, or the cassette tape.

PUTGET can buffer up to two files: one of which must have been opened for input or update, the other for output. Any further files are passed straight through to the NFS and will not be buffered.

PUTGET only allows access to files on a sequential basis, that is there is no facility for using PTR# and EXT# although EOF# of course still works. If PTR# and EXT# need to be used, or if more memory is required (PUTGET takes up 3 pages of RAM, i.e. it raises the BASIC variable PAGE by &300) then PUTGET must be de-activated. To do this type either <Ctrl-Break> or \*FX247<Return> then <Break>.

In detail, what happens is that PUTGET is a short loader program that executes at &A00. This loads the file PRL.PUTGET, which is the actual PUTGET code in relocatable format (this must be in the library in the sub-directory PRL) at the operating system 'high water mark' (OSHWM), and then relocates and executes it. This code intercepts various vectors in page 2 (OSARGS to OSFSC and OSBYTE) and the BREAK key intercept, implodes the font (using \*FX20,0) and raises OSHWM by 3 pages. It will then re-explode the font and finally re-start the current language (which in the case of BASIC is needed to re-initialise PAGE to the new value of OSHWM).

## Examples:

\*PUTGET

Ideally this should be run at the beginning of every session, perhaps by placing it in a !BOOT file, either for the user, or in the library. See under \*OPT4 for details.

## **Likely Errors:**

There are no error messages specific to PUTGET, except for the non-fatal **Putget already active**. PUTGET will produce this message and cause no further action.

PUTGET loads a file called PRL.PUTGET from the library, and that the error **PUTGET not found** or **PRL not found** will occur if this file or directory PRL are absent.

Note that running PUTGET after LOADING a BASIC program will give nothing if the user attempts to list or run the program.

## **Associated Keywords:**

BGET, BPUT

## **Compatibility Notes**

This utility will run identically with Acorn File Servers, and its use is recommended with them as well as with SJ Research File Servers.

---

**Syntax:** \*REMOTE <station number> | \*REMOTE <User Id>

## Action with Wild Cards in the User Identifier:

Occurs on the first match, from the top of the list as produced by \*USERS (see this Section) downwards.

## Description

This utility allows the user to take over one other specified station, or the station where a specified user is logged-on. This station will then echo the screen output from your station. This is useful for demonstrations, but it is a good idea to check that this will not disrupt the other user in the middle of an operation.

Control of a remote station is turned off by the command \*ROFF, which is not kept in the utilities library but is provided in the network ROM inside the machine. Attempts to take over a second station at the same time will have no effect and it is not possible to take over your own station.

The commands \*PROT and \*PROTEX can be used to protect your machine against \*REMOTE. In that case, or if a station number that is not logged-on is specified, the \*REMOTE machine will wait until <Escape> is pressed.

If the specified user is logged-on at several stations, only one of them will be taken over: this will be the station at which the specified user last performed a filing operation, and also the first one that would appear in the list produced by the program \*USERS (see this Section).

## Likely Errors:

**Not logged on**

**Error 174 (AE)**

If a user identifier was specified, and that user was not logged on at the currently selected File Server.

## Associated Keywords:

\*NOTIFY, \*PROT, \*ROFF

## Compatibility Notes

Supported by Acorn systems.

---

**Syntax:** \*RENAME <old general specifier> <new general specifier>

## Action with Wild Cards in File Name:

Occurs on every match. See below for full details.

## Description

This command changes the name of an existing file or directory. The user must own the item (i.e. he must have access to the main or auxiliary account of the item). An attempt to rename an item so that it would have the same name as an existing file (or directory) will give the error message **Already exists**.

A file (or directory) may be moved into another directory using this command, but the user must be an owner of the new directory as well as of the item being renamed (even if he is renaming the item solely within one directory). Note that he does not have to be an owner of the directory from which he is moving a file.

An attempt to rename a directory as a sub-directory of itself will give rise to the error **FS unusual error 07**.

In a system with more than one disc, it is not permitted to use this command to move files from one disc to another. (The utility programs COPIER or MULTICOPY should be used for this).

Wild card characters are permitted in the file specifiers, and will cause the renaming to occur on all items that match the wild card specifier. The same wild card character(s) must appear in the old and new specifiers.

## Examples:

```
*RENAME MYPROG $.PROJECT.MYPROG
```

will move MYPROG into directory \$.PROJECT without changing its name. The user must be an owner of \$.PROJECT for this to be allowed.

```
*RENAME PROG* DATA*
```

will rename PROG1 as DATA1, PROGDEMO as DATADEMO and so on.

```
*RENAME A#PP* DATA#PP*
```

will rename AZPP1 to DATAZPP1, AMPP to DATAMPP and so on, but not A1PPINFO to DATA1PPINFO, because the new name has more than 10 characters and so would cause an error -- under the wild card rules, this item would be passed over (see Section 6.3). Note also that the wild cards must correspond in the old and new specifiers.

## Likely Errors:

**Circular RENAME**                      **Error 7 (07)**

Caused by an attempt to rename a directory as a sub-directory of itself.

**Renaming across discs**              **Error 176 (B0)**

It is not possible to transfer an item from one disc to another using \*RENAME.

**Insufficient access**                  **Error 189 (BD)**

The user does not own either the file being renamed, or the directory into which he is trying to rename it.

**Bad wildcard****Error 204 (CC)**

The number and types of wild card characters must match in the \*RENAME command.

**Already exists****Error 196 (C4)**

From an attempt to rename an item to have the same name as an existing item.

## Compatibility Notes

Supported by Acorn systems. Note that you may *not* rename directories on an Acorn File Server, and that wild cards are not allowed.

---

**Syntax:** \*REROUTE <print job name> <logical printer name>

## Action with Wild Cards in File Name:

Occurs on every match.

## Description

This command changes the logical printer selected for a print job in the print queue on the currently selected File Server. The user must have owner access to the print job and be allowed to use the new selection for the logical printer. There is no need to give the full pathname of the print job; this command will automatically look in %PRINTQ since it can only apply to files there.

\*REROUTE is particular useful in combination with the special logical printer HOLD; as output can be routed to HOLD, inspected to see that it is as required, and then sent to another logical printer with \*REROUTE.

## Examples:

\*REROUTE AA03 fancy

will change the logical printer selected for the print job AA03 to the printer fancy.

## Likely Errors:

### FS Unusual Error 46

An attempt to reroute a file or directory, not a print job.

### Not authorised to use printer Error 10 (0A)

If the system manager has restricted the use of this logical printer to users with access to a certain account.

### Insufficient access Error 189 (BD)

Only an owner can change the logical printer for a print job.

### xxxx not found Error 214 (D6)

If either the print job or the logical printer specified does not exist.

## Associated Keywords:

\*PS, \*PRINTOUT, \*PRINTER

## Compatibility Notes

Not supported on Acorn systems.

---

**Syntax:** **\*RUN** <file specifier> | \*/<file specifier> | \*<file specifier>

## Action with Wild Cards in File Name:

Occurs on first match (alphabetically).

## Description

This command will load and run a machine code program at a location specified when the file was saved (see under \*SAVE).

The action taken by the File Server is to search through the currently selected directory for the file specified. If it is not found, the File Server will search through the currently selected library (see under \*LIB command), and will load the file from there if found.

The command \*/<file specifier> is exactly equivalent to \*RUN <file specifier>. If there is no sideways ROM or filing system command with the same name as <file specifier> then the version \*<file specifier> may be used. If there is such a command, then the full version \*RUN (or \*/) must be used.

## Examples:

\*PUTGET

runs the program PUTGET from the currently selected directory (CSD) or, if it is not found in the CSD, from the library.

\*RUN DELETE

will run a program called DELETE from the CSD or library. This form of command avoids the File Server command \*DELETE.

\*/DELETE

is identical to the example directly above.

## Likely Errors:

xxxx is not a file                      **Error 181 (B5)**  
An attempt has been made to \*RUN a directory.

Insufficient access                    **Error 189 (BD)**  
The user must have access R to the file.

xxxx not found                        **Error 214 (D6)**  
The file does not exist.

## Compatibility Notes

Supported on Acorn systems.

---

**Syntax:** **SAVE** "<file specifier>" | **SAVE** <string variable>

## Action with Wild Cards in File Name:

Wild cards prohibited.

## Description

SAVE is a BASIC command, that causes the BASIC program currently in memory to be saved, with a file-name equal to the string immediately following the SAVE command.

The user should take care to type **OLD** after pressing <Break> on the BBC Microcomputer. If the program is saved after <Break> without doing this, then a trivial file of two bytes length will be created, over-writing any existing file of that name.

The system manager has an option available to prevent users SAVEing files of less than 16 bytes in length, to prevent the above problem. An attempt to save a short file will give the error **Too short**.

## Examples:

```
SAVE "Prog27"
```

will save the current program as **Prog27**.

```
A$="MyProg"
```

followed by:

```
SAVE A$
```

will save the current program as **MyProg**.

## Likely Errors:

### Too short

#### Error 6 (06)

Caused usually by doing SAVE after pressing <Break>. The system manager has to set a special option before this error will be caused, and its occurrence can be changed using \*ENABLE SAVES or \*DISABLE SAVES.

### xxxx is a directory

#### Error 181 (B5)

Caused by attempting to create a file with the same name as that of a directory.

### Insufficient access

#### Error 189 (BD)

Caused if the user is not an owner (i.e. does not have access to the main or auxiliary account) of the directory in which the new file is to be created.

### Already opened by xxxx

#### Error 194 (C2)

If this or another user has this file open for reading or writing, then it cannot be deleted by creating a file with the same name, until it has been closed.

### Bad name

#### Error 204 (CC)

If the file name contains illegal characters (e.g. \$ % . ^: except in contexts where they are permitted)

### Bad wildcard

#### Error 204 (CC)

Use of \* or # in the file name.

**Entry locked****Error 195 (C3)**

It will be necessary to use the \*ACCESS command to unlock the file before it can be deleted by creating a file with the same name.

**Associated Keywords:**

LOAD, \*SAVE

**Compatibility Notes**

Supported on Acorn systems.

**Syntax:** **\*SAVE** <file specifier> <base addr.> <top addr.> |  
**\*SAVE** <file specifier> <base addr.> <top addr.> <execute addr.> [<load addr.>] |  
**\*SAVE** <file specifier> <base addr.> + <length> |  
**\*SAVE** <file specifier> <base addr.> + <length> <execute addr.> [<load addr.>]

## Action with Wild Cards in File Name:

Wild cards prohibited.

## Description

\*SAVE creates a new file, containing a copy of the specified memory area.

The base address is the address in memory from which the copying will start, and the top address the address *next above* the last one that will be copied. If the length is specified after a + sign, the top address will be found by adding the length to the base address.

When a file is saved, the values of the load address and the execute address are stored in the appropriate directory. The execute address is the address in memory that the computer will jump to after loading the file in a \*RUN command, and the load address is the base address for the file when loading with a \*RUN or \*LOAD command.

If the load address is not specified, then it will be taken as equal to the base address. If the execute address is not specified, it will also be taken as equal to the base address. If you wish to specify the load address you *must* also specify the execute address.

## Examples:

\*SAVE IMAGE 2000 280E

will save the area of memory between 2000 and 280D (inclusive) into a file called IMAGE.

\*SAVE COMP 2000 295E 3020 3000

will save between 2000 and 295D into a file called COMP. When \*COMP is typed, the file will be loaded at address 3000, and the system will then jump to address 3020.

## Likely Errors:

### Too short

#### Error 6 (06)

Caused by an attempt to save a file of less than 16 bytes length. This error is caused if the system manager has set a special option for a user, and its occurrence can be changed using \*ENABLE SAVES or \*DISABLE SAVES.

### xxxx is a directory

#### Error 181 (B5)

Caused by attempting to create a file with the same name as that of a directory.

### Insufficient access

#### Error 189 (BD)

Caused if the user is not an owner (i.e. does not have access to the main or auxiliary account) of the directory in which the new file is to be created.

### Already opened by xxxx

#### Error 194 (C2)

If this or another user has this file open for reading or writing, then it cannot be deleted by creating a file with the same name, until it has been closed.

**Bad name****Error 204 (CC)**

If the file name contains illegal characters (e.g. \$ % . ^: except in contexts where they are permitted)

**Bad wildcard****Error 204 (CC)**

Use of \* or # in the file name.

**Entry locked****Error 195 (C3)**

It will be necessary to use the \*ACCESS command to unlock the file before it can be deleted by creating a file with the same name.

**Associated Keywords:**

\*LOAD, SAVE

**Compatibility Notes**

Supported on Acorn systems.

---

**Syntax:** \*SDISC <disc name>

## Action with Wild Cards in Disc Name:

Occurs on first match in order of disc drive numbers.

## Description

This command is equivalent to \*DIR:<disc name> Note that the subsequent use of \*DIR (without an argument) will return to the initially selected disc.

## Examples:

\*SDISC MAIN2

will select as CSD the root on disc MAIN2.

## Likely Errors:

**Bad name**                                      **Error 204 (CC)**  
If the disc name contains illegal characters.

**xxxx not found**                              **Error 214 (D6)**  
If the disc of that name is not present in the system.

## Associated Keywords:

\*DIR

## Compatibility Notes

Supported in Acorn systems, but note that \*SDISC is equivalent to \*DIR:<disc name>.<user identifier>, and that a subsequent \*DIR will remain on the new disc on Acorn File Servers.

**Syntax:** \*SPOOL <file specifier> | \*SPOOL

## Action with Wild Cards in File Name:

Wild cards prohibited.

## Description

This command causes the specified file to be created (deleting any existing file of the same name), and for all subsequent output to the screen of the BBC Microcomputer to be written to it. A common application is to make text files out of BASIC programs, or as a simple way of making program generated text. Some examples are given below.

The file is closed by running \*SPOOL without a file specifier. Note that the file handle in the BBC Microcomputer will be lost if <Break> is pressed, and so the spooling will stop. The File Server will still have the file open however, so under these circumstances the file will have to be closed either with CLOSE#0 or by logging off with \*BYE.

In detail, \*SPOOL performs an OPENOUT operation, and then uses the BPUT call to send all text from the BBC Microcomputer screen to the specified file. This operation is fairly slow over the Econet network, so it is recommended that the utility program \*PUTGET is run before using \*SPOOL (see description under \*PUTGET)

## Examples:

This type of program allows the user to create a file !BOOT:

```
10 *SPOOL !BOOT
20 PRINT "*TV0,1"
30 PRINT "*| THIS IS THE MESSAGE OF THE DAY"
40 PRINT "*PUTGET"
50 *SPOOL
```

The sequence below will produce text from a BASIC program, which can then be edited by a word processor. The \*EXEC command (see this Section) can be used to turn the edited text back into a BASIC program:

```
*SPOOL TEXTFILE
LIST
.
.
.
*SPOOL
```

## Likely Errors:

**xxxx is not a file**                      **Error 181 (B5)**  
Caused by attempting to create a file with the same name as that of a directory.

**Insufficient access**                      **Error 189 (BD)**  
Caused if the user is not an owner (i.e. does not have access to the main or auxiliary account) of the directory in which the new file is to be created.

**Too many files open**                      **Error 192 (C0)**  
There is a limit to the number of channels available, normally 5 or 6.

**Already opened by xxxx      Error 194 (C2)**

If this or another user has this file open for reading or writing, then it cannot be deleted by creating a file with the same name, until it has been closed.

**Bad name      Error 204 (CC)**

If the file name contains illegal characters (e.g. \$ % . ^: except in contexts where they are permitted)

**Entry locked      Error 195 (C3)**

It will be necessary to use the \*ACCESS command to unlock the file before it can be deleted by creating a file with the same name.

**Associated Keywords:**

\*EXEC, \*PRINTOUT

**Compatibility Notes**

Supported on Acorn systems

## Syntax: \*STATEMENT

### Description

This program gives a list of all the accounts that the user has access, with the associated credit balances.

The credit balance of an account is in units of 1 kilobyte (K), and represents the space available for files (or directories) with the corresponding account number. If a user has access to more than one account, he may move files from one account to another by use of the \*ACCOUNT command (see this Section). The account of a file or directory also determines the ownership of that item -- if a user has access to the account of an item, then he owns that item. A full explanation is given in Section 5.4.

If there is more than one disc in the system, then \*STATEMENT will give a separate list of accounts for each disc. The user will however only be able to create files on discs *on which there exists a directory with account to which he has access*.

Transient programs run in a reserved area of workspace in the computer, and will not corrupt programs in the main memory.

### Examples:

\*STATEMENT

The system will reply (for example):

```
Disc 0
Account  Balance
  88      252k
  89       66k
  90     bankrupt
```

```
Disc 1
Account  Balance
  88       45k
  89     512k
  90     511k
```

### Likely Errors:

There are no errors specific to this program.

### Associated Keywords:

\*ACCOUNT, \*CREDIT, \*DEBIT, EDITPASS

### Compatibility Notes

Not supported on Acorn systems, since these do not run an accounting system.

---

**Syntax:** \*STATIONS [<network number>]

## Action with Wild Cards in Network Number:

Wild cards prohibited.

## Description

This utility displays a list of all machines connected to the Econet network, that are currently switched on. Each line is of the form:

<station number> <machine type> <version number>

The machine types include:

- Acom BBC micro
- Acom Master
- Acom E.T.
- Acom System 5
- Acom System 3/4
- SJ Research File Server
- SJ Research Z80 CP/M
- SJ Research IBM
- SJ Research SCSI

\*STATIONS will *not* show the number of the computer running this program (the program \*CV will do this if required). It will also not show the numbers of any computers in which the network interface has been disabled -- the program NETMON and some computer games do this.

If the installation has *multiple networks connected by bridges*, then \*STATIONS with no argument will display only active stations on the same network. If \*STATIONS is followed by a number between 1 and 255, then the program will look for a bridge joining to the network of this number, and will list the stations active on that network.

## Examples:

\*STATIONS

to which the system will reply (for example):

Station	Type
254	SJ Research File Server
135	SJ Research SCSI
87	Acorn BBC micro
32	SJ Research Z80 CP/M
8	Acorn Master
2	Acorn BBC micro

\*STATIONS 3

will search for network number 3, and will produce a list similar to above, of stations on network 3.

### **Likely Errors:**

There are no errors specific to this program.

### **Associated Keywords:**

\*CV, \*FSLIST, \*PSLIST

### **Compatibility Notes**

Supported by Acorn systems.

---

## Syntax: \*TIME

## Description

This utility prints out the time and date, from the a 'real time clock' contained to the File Server. The form of the output is given in the example below.

There are alternative versions of this program suitable for incorporating into user programs. These are **\*PTIME** **\*PDATE** **\*PDATE2** -- all three produce output without any <Return> characters around the text. See under **\*PTIME** for details. There is also a program **\*GTIME** which inserts the time in a machine readable form into the BBC Microcomputer.

## Examples:

\*TIME

The system will reply (for example)

The time is 11:10:46 on 08/03/84

## Likely Errors:

There are no errors specific to this program. A nonsense time output implies that the real-time clock has not been set, or that the File Server has been switched off for several months, so that the clock battery has run down. Ask your system manager for help.

## Associated Keywords:

\*GTIME, \*PTIME, \*PDATE, SETTIME

## Compatibility Notes

Supported on Acorn systems. Level 2 File Servers have only the date; level 3 File Servers have both date and time, and they use the interval timer function in the BBC Microcomputer thereafter -- there exists the possibility that this timer could lose time under certain circumstances.

---

**Syntax:** \*TYPE <file specifier>

## Action with Wild Cards in File Name:

Occurs on first match (alphabetically).

## Description

This program opens the specified file, and prints it as a text file on the screen of the computer. Tab characters (ASCII value 9) will be replaced by sufficient spaces to bring the next character to a multiple of eight character spaces from the left margin.

Note that if the file contains 'control' characters (ASCII characters 0-31), then these not be printed to the screen. (This is because they would affect the screen directly, and probably make a nonsense of the output -- for example ASCII 21 <Ctrl-U> will turn the screen off altogether). The only exceptions to this are the characters &09 (tab), &0C (page feed) and &0D (new line), which will be printed. Characters of ASCII value greater than &80 ('top bit set' characters) will also not be printed.

This program uses the multiple byte transfer OSGBPB call, and so will run considerably faster than for example the version of \*TYPE that is contained in the DFS ROM. If DFS is fitted to the computer, use \*/TYPE to be sure of running the network version. (See under \*RUN for full details of \*/).

## Examples:

\*TYPE FILE1

will print FILE1 to the BBC Microcomputer screen. A printed copy could be made at the same time by typing <Ctrl-B> before the command, to turn on the printer, and <Ctrl-C> at the end to turn it off. See Sections 5.5 and 6.5 about printing through the network.

## Likely Errors:

There are no errors specific to this program. However, it performs an OPENIN call, and so can cause all the same errors that the OPENIN would.

## Associated Keywords:

\*DUMP

## Compatibility Notes

Supported by Acorn systems.

**Syntax:** \*UNPROT

## **Description**

This program removes the protection that has been set up by \*PROT from the station at which it is entered. \*PROT prevents other stations sending operations which affect your station. Protection will also be removed by a <Ctrl-Break>.

If some operations from other stations are required but not others, an extended version of this command is given by \*PROTEX.

## **Likely Errors:**

There are no errors specific to this command.

## **Associated Keywords:**

\*PROT, \*PROTEX

## **Compatibility Notes**

Supported by Acorn systems.

## Syntax: \*USERS

### Description

This utility prints a list of all the users currently logged-on to the File Server, the station numbers at which they are logged-on, and any system privilege that they may have.

If a user is logged on at several stations, then he will appear several times in the list. The list is re-ordered every time a filing system operation occurs, so that the user who most recently performed a filing operation will be at the top of the list -- this will, of course, be the user who ran \*USERS from the network.

It is worth noting that a command like \*NOTIFY <user identifier> will only notify one occurrence of the user. The one which will be notified is the same as the one that appears first in the list produced by this program, which should be the last computer at which the specified user performed a filing operation.

Note that if a user was logged-on, and switches off his machine without typing \*BYE, he will still appear in \*USERS until someone else turns that machine back on and logs-on from it.

The users listed may include \*-SYSTEM-\* and \*-SPOOL-\* at station 0. These are used by the system to carry out print spooling and other system operations. These users do not have system privilege.

### Examples:

\*USERS

to which the system will reply (for example):

Station	User Id.	Privilege
026	SYST	System
043	FRED	
000	*SYSTEM*	
218	JOHN	

### Likely Errors:

There are no errors specific to this program.

### Associated Keywords:

\*CV, \*GUSERS, \*PUSERS, \*STATIONS

### Compatibility Notes

Supported on Acorn systems, but the user list is not sorted in the same way, and is essentially random in order.

## **Syntax: \*VERS**

## **Description**

This program displays the current version number of the File Server. This will be of the form:

**SJ Research File Server ver n.nn / <File Server type>**

## **Examples:**

\*VERS

to which a typical reply will be:

SJ Research File Server ver 0.67/HDFS

for a hard disc File Server.

## **Likely Errors:**

There are no errors specific to this program.

## **Associated Keywords:**

\*FSLIST

## **Compatibility Notes**

Supported by Acorn systems, but the displayed number will be different.

---

**Syntax:** **\*VIEW** <station number> | <user identifier>

### Action with Wild Cards in the User Identifier:

Occurs on first match (chronologically).

### Description

This utility allows a station to make a complete copy of a remote station's screen. The prompt will then be returned.

### Examples:

**\*VIEW FRED**

Will copy the screen of the station number that FRED is logged on to. If there is more than one occurrence of FRED in the user list, then the station which he most recently used to access the File Server will be copied.

**\*VIEW 1.235**

will copy the screen of station number 235 on network number 1.

If the remote machine is in a screen mode which takes more memory than the screen mode in the local machine then the error 'Mode x' will be returned, where x is the screen mode of the remote machine. After the 'Mode x' error the value of x can be read by OSWORD A=&13 with function code 10. For example:

```
10 DIM S% 15
20 S%?0=10
30 X%=S%:Y%=S% DIV 256
40 A%=&13:CALL &FFF1
50 PRINT"Screen mode is ";S%?1
```

### Likely Errors:

**Inactive** **Error &A2**

If the remote station is not on the network.

**Mode?** **Error &A4 Mode**

If the remote station is a fileserver or some other machine with a different screen layout.

**PROT** **Error &A2**

If the remote station has issued a \*PROT command.

**Not logged on** **Error &AE**

If the user identifier specified is not currently logged on.

**Mode x** **Error &AD**

If the screen mode of the remote station uses more memory than the screen mode you are currently in.

### Associated Keywords:

\*PROT, \*REMOTE

## Compatibility Notes

Supported by Acorn systems.

# Chapter 4: An Introduction to System Management

---

## 4.1 Operating instructions for the MDFS

SJ Research File Servers are designed to be run easily without the need for much specialist knowledge. This section explains how the system manager should get going, setting up the communal filing system for all his users.

This chapter is kept deliberately simple, in order to explain the principles. System managers will find full descriptions of all the commands and utility programs in Chapters 5 (user management) and 6 (printer management).

If the File Server has not been installed for you please refer to Appendix C which gives full instructions.

### 4.1.1 Switching on the File Server

The Modular Disc File Server needs to be switched on at the front panel key-switch; the green POWER light will come on. All the lights, except the red NO CLOCK light, will come on for two or three seconds while the hardware is tested (consult Appendix B if the system failure light flashes). For normal operation, the switch should be turned to the SECURE position. The yellow DISCS FREE light will flash after a few seconds -- now insert all the floppy discs that you wish to use. There must be a copy File Server program on one of the discs at this stage, but it may be on either a floppy disc or a hard disc; there is a copy on the disc supplied with the fileserver. Press the RELEASE DISCS button on the front panel, and the DISCS FREE light will flash more rapidly while the system loads the program. After about ten seconds, the green ON LINE light will come on and the DISCS FREE light will go out.

The fileserver is now ready for use. If you want to perform any system privileged user operations (for example editing the password file or setting the internal clock), then you should now turn the switch to the SYST position. More fundamental system operations such as formatting discs are carried out in Utility Mode, which is reached by either starting up the fileserver as above, but with the key-switch in the SYST position from the beginning, or by typing \*FINISH from a station on the network while the fileserver is running normally.

If, after the File Server is switched on, the RELEASE DISCS button is not pushed within thirty seconds the system will attempt to start the File Server automatically.

### 4.1.2 Switching off the File Server

Before switching off the File Server, press the RELEASE DISCS button on the front panel. After a short pause the yellow DISCS FREE light will flash alternately with the green ON LINE light. Remove all the floppy discs, and turn the front panel key-switch to the STOP position -- the front panel lights will extinguish.

If the system is in the process of printing, it will stop when the RELEASE DISCS button is pressed; when the File Server is turned on again, previous print requests will be remembered and restarted, going back to the beginning of any uncompleted item in the print queue.

*Do not switch off without pressing the RELEASE DISCS button, for example using a master switch. (If a master switch is in use, then the File Server should run from a supply independent of it). If the system is not stopped before turning off, the cache memory may not have been written back to the discs. In this case, it will probably be necessary to restore the system from a backup, as the data on the discs will be corrupted. It is also wise to remove any floppy discs from the drives before turning off the File Server, as power-off may*

have unpredictable effects on them.

If hard discs are fitted, we recommend not switching off the File Server at all, since hard discs are considerably more reliable if they are kept spinning continuously at all times. If your local Fire and Safety regulations permit it, this is the best course.

### 4.1.3 Changing Discs

You may want to run a File Server system where there are more floppy discs (excluding backups) than floppy disc drives, and you insert them perhaps when a particular class is present. This process is recommended only when your Econet network extends only over a single room -- beyond that you will not normally know who will want access to files at any time, so the only complete solution is to have a File Server system large enough to have all files on line at all times.

The following procedure is necessary to change discs, either for this purpose, or any other (e.g. to return to an earlier version of a disc).

Press the RELEASE DISCS button on the front panel. The File Server may continue to access the discs for a few seconds, in order to write back any information held in memory. When it has finished, the DISCS FREE light will flash alternately with the ON LINE light. While the system is in this state, no File Server discs can be accessed by users. Remove the old disc(s), insert the new disc(s), and press the button again. Note that it is *not* necessary to continue using the disc containing the fileserver program once the system is running : the program is retained in memory.

When the discs are changed, the File Server will discard the old user list for the changed discs (since the password file has probably changed). This means that all users who are logged on from those lists will have to log on again -- otherwise they will receive the error message **Who are you ?** if they attempt a filing operation. All opened files on changed discs will be closed. Discs which have not been changed will be unaffected.

*Do not change discs without pressing RELEASE DISCS and then waiting for the DISCS FREE light.* The system keeps information about the disc data, and transient information, in memory. If you change a disc without telling the system, the vital root directory information will be corrupted, and it will almost certainly be impossible to access files on the affected discs afterwards.

## 4.2 Organising the File Server

Your system will have a number of different users to whom you will want to be able to give facilities to create files for themselves, to read certain communal files (for example library programs) and to have selective access to other users' files.

The list of authorised users in a SJ Research File Server is kept in a file called the *password file*. This file can be read and saved only by someone with *system privilege* -- normally only the system manager himself and only when the front panel key-switch is turned to the SYST position. The password file contains information about each user: his password, any accounts he has access to, and administrative information concerning start-up (boot) options, library directories and user root directories.

If someone logs-on to the system, and his name does not appear in the password file, then he will be logged on as the *default user*, if one has been set up by the system manager using EDITPASS (see Section 4.3). If no default user has been set by the system manager, the user will receive the message **User not known**.

When a user listed in the password file logs-on, any password he quotes will be checked against the one in the password file, before the log-on is allowed to proceed. He will then be given any rights and privileges listed against his name in the password file. The system will then search the disc on which the user's password file entry was found, for the *User Root Directory* specified for that user in the password file, which by default has *that user's name*, and will set this to be the currently selected directory (see Section 3.3 under \*I AM for details). If no appropriately named directory is found, the disc root directory will be selected.

As described more fully in Section 3.3 (under \*ACCESS and \*ACCOUNT), the *account(s)* to which a user is given access, control two things:

First, every file (or directory) has an account number, and *if a user has access to this account, then he is an owner of that file (or directory)*. Only an owner may create files in a directory, and only an owner may delete a file or change its access letters (see Section 3.3 under the \*ACCESS command). Note that there can be *more than one owner* of a file (or directory), simply by allocating access to its account to more than one user -- this can be useful for communal files in a project.

Second, each account has a *credit balance* of storage space, and an attempt to create a file which would cause that balance to become less than zero will be prevented, and cause the error **Account bankrupt**.

## 4.3 Command details

This section describes the operation of each of the commands and programs intended for use by the system manager for the day to day running of the File Server system. Chapters 5, 6 and 7 explain in more detail how these commands are used for particular purposes.

These commands are not in themselves destructive in the wrong hands: provided that the system manager's password is kept secure, no other user can actually change any settings, although some of the programs allow non-sensitive information to be read. In addition, the front panel key switch must be in the SYST position to enable the most sensitive operations.

The commands and programs documented here are:

*CHECK	Machine code program
*CREDIT	File Server command
*DEBIT	File Server command
EDITPASS	BASIC program
EDITPRINT	BASIC program
*FAST	Machine code program
*FINISH	File Server command
FORCER	BASIC program
LISTUSERS	BASIC program
*LOGOFF	File Server command
*PGO	File Server command
*PSTOP	File Server command
SETTIME	BASIC program
SIZER	BASIC program

## Syntax : \*CHECK

## Description :

This program will check the issued library utility programs using a 'checksum' algorithm, and will compare them with internally held values. The program will also print whether PSnofx6 or PSFX6 has been renamed as PS. It contains its own list of files to check, and it will list as it runs, and print a warning message if any of them have been tampered with.

Each file will be listed, with no further comment if its checksum was correct. If it was not correct then either the message **is old** or **is corrupted** will be displayed. If the file was not found, **doesn't exist** will appear.

If the currently selected directory is likely to contain any files of the same name as these utilities, then it is wise to type **\*DIR \$.LIBRARY** before running CHECK.

CHECK is encrypted to make it difficult for casual hackers to modify it, and it performs a checksum upon itself when run, but system managers are recommended to keep it on DFS floppy disc, locked away, and perhaps not leave it anywhere in the File Server.

CHECK will be re-issued with any new version of the library. If the new version of CHECK is run on the old version of the library then it can be used to find out which utilities have changed.

## Example :

**\*CHECK**

Utility CHECKer program version 2.nn

BUILD is corrupted

CLOSE

COPIER is old

CV doesn't exist

.

.

PS with \*FX6

.

.

.

Having set up the system for the first time you will need to decide whether your printer needs a **\*FX6** or not. So rename it **PSnofx6** if you don't by:

**\*RENAME \$.LIBRARY.PS \$.LIBRARY.PSfx6**

**\*RENAME \$.LIBRARY.PSnofx6 \$.LIBRARY.PS**

## Compatibility Notes:

Not supported by Acorn systems, as it is specific to SJ utility programs.

---

**Syntax:** \*CREDIT <account number> <amount>

## Action with Wild Cards:

Wild cards prohibited.

## Description:

This command adds the specified <amount> in units of 1 kilobyte (K) to the balance of account <account number>. Account numbers range between 0 and 3FF and are in hexadecimal, but the amount of credit is in decimal.

The maximum balance possible is 65535K, and the system manager may effectively turn off the space accounting system by setting accounts this much credit: 65536K is the largest single disc that can be fitted to a system.

System privilege is required to execute this command.

The accounts system gives control of disc space filled by users' files. This command increases the allocation of space available for files with the specified account number.

Note that the current credit balance of an account represents only the free space available for files, and does *not* include files already existing with that account. It is therefore wise for the system manager to keep a notebook containing details of users, their accounts and any subsequent credit or debit given to them. A utility program **SIZER** is provided to find the total size of files in a directory tree, and this could be used to find the total space taken up by a user's files at a later date.

The transient program **\*STATEMENT** will give a printout of all the accounts to which a user has access, with the associated credit balance for each. The system manager will normally have access to all accounts, and so will get a long list of all accounts from 0 to 3FF for each disc in the system. It may be wise to send this output to the printer if all the information is required.

## Example:

```
*CREDIT 89 512
```

allocates a further 512K to account number 89.

## Likely Errors:

**Bad number** **Error 4 (04)**

If there is not a number specified in both fields or the number is in the wrong base (hex for the account number, decimal for the amount).

**Key Locked** **Error 5 (05)**

If the key switch on the front panel of the MDFS is not in the SYST position.

**Insufficient privilege** **Error 186 (BA)**

If the user does not have system privilege.

## Compatibility Notes:

Not supported by Acorn systems.

---

**Syntax:** \*DEBIT <account number> <amount>

## Action with Wild Cards:

Wild cards prohibited.

## Description:

This command subtracts the specified <amount> in units of 1 kilobyte (K) from the balance of account <account number>. Account numbers range between 0 and 3FF (hexadecimal), but the amount to be debited is given in decimal.

The maximum balance possible is 65535K, so the system manager may wipe out the balance of an account by typing :

```
*DEBIT <account number> 65535
```

System privilege is required to execute this command.

The accounts system allows control of disc space filled by users' files. This command decreases the allocation of space available for files with the specified account number.

Note that the current credit balance of an account represents only the free space available for files, and does *not* include files already existing with that account. It is therefore wise for the system manager to keep a notebook containing details of users, their accounts and any subsequent credit or debit given to them. A utility program **SIZER** is provided to find the total size of files in a directory tree, and this could be used to find the total space taken up by a user's files at a later date.

The transient program **\*STATEMENT** will give a printout of all the accounts to which a user has access, with the associated credit balance for each. The system manager will normally have access to all accounts, and so will get a long list of all accounts from 0 to 3FF for each disc in the system. It may be wise to send this output to the printer if all the information is required.

## Example:

```
*DEBIT 89 512
```

removes 512K from account number 89.

## Likely Errors:

**Bad number** **Error 4 (04)**

If there is not a number specified in both fields or the number is in the wrong base (hex for the account number, decimal for the amount).

**Key Locked** **Error 5 (05)**

If the key switch on the front panel of the MDFS is not in the SYST position.

**Insufficient privilege** **Error 186 (BA)**

If the user does not have system privilege.

## Compatibility Notes:

Not supported by Acorn systems.

---

## Syntax: CHAIN "EDITPASS"

### Description:

The system maintains a special file %PASSWORDS, which does not exist in any particular directory, and is only 'visible' to system privileged users. The password file cannot be deleted with \*DELETE, but can only be cleared in Utility Mode, although it is possible to save over the top of it. The name must be quoted in full in upper case, i.e. no wildcards. EDITPASS is provided in the library for editing the contents of %PASSWORDS.

EDITPASS is a screen editor for editing password files, or for creating new ones. The present version can handle about 200 users, and requires a BBC computer with 32K RAM (i.e. Model B or expanded Model A). If a BBC with a 6502 second processor is used, more than 300 users can be created.

(There is a version of this program, EDITPASS!, which uses more readable identifiers, if the system manager wants to see how the program works or to make his own version. EDITPASS itself has been condensed, so as to leave maximum storage for data).

Although anyone can run EDITPASS (assuming that the system manager has set public read access to this file), it is necessary to have system privilege *and access to account 0*, and the MDFS front-panel keyswitch must be in the "SYST" position to either read or save the password file on a disc.

EDITPASS always works with the currently selected disc drive: to edit the password file on another drive, use \*DIR :<disc name> to select this new disc.

*Be cautious when running this program. All the system passwords will be loaded into memory, and may be displayed on the screen. Never walk away from the computer running EDITPASS without typing \*BYE and switching off the power.*

When the program is run, the program will prompt:

```
Password file Editor 27feb86
Maximum number of users=257
Options:
  E - edit PW file from disc
  N - create new PW file
  O - edit file from current RAM
?
```

The normal option is E, unless it has been necessary to delete the password file for some reason. The O option is useful if this program has stopped, either with an error, or as a result of pressing the <Escape> or Q key.

The program will then display all the users, with their boot options and system privilege (if any). The display will be similar to:

A(dd) D(elete) S(ave) X(pand) Q(uit) \*

User id	Boot option	Privilege
ARG	0 Off	
BASHER	0 Off	System
BJ	0 Off	
BOOT	3 Exec	
CLAIRE	0 Off	
DEFAULT	2 Run	*DEFAULT*
JEF	0 Off	
JOHN	3 Exec	
KIM	2 Run	

The 'up' and 'down' cursor keys scroll the display, allowing the total user list to be available. To add new users, press the A key: the program will prompt repeatedly for names until the list is terminated with <Return> on its own.

D key will delete the user on the current line.

\* key allows normal \* commands to be run from within the program. To return to the list of users, type <Return> on its own on a line.

Q key stops the program without saving the result back to disc. The <Escape> key, pressed at any stage, has the same effect, except in the \* mode.

S key saves the password file to disc, and stops. A check will be made that there is at least one system privileged user, and that a user exists with access to account 0 (these are both vital to the running of the system), and the program will print a warning if one of these requirements is not met. Note that you may however want to keep all your system users on one (or a few) discs for security, in which case it would be legitimate for there to be no system user, or user with access to account 0, on other discs. The name of the *default user* if any (see below) will be displayed, and then the prompt Save (Y/N) : Typing N will return to the list of users.

*After typing Q or S, at the end of the system manager's session, type \*BYE then TURN OFF THE BBC MICROCOMPUTER at which you were logged on. The password file will remain in the computer unless it has been overwritten, and another user could easily read it from there, and gain unauthorised access to the system.*

X key enters the *expanded mode* for the user at the current cursor position. The display will change to, for example:

```
A(ccounts) S(ystem priv) D(efault user)
P(assword) U(RD) L(IB) 0..3 Boot option
8-Saves 9-PW lock E(nable)

ARG          0 Off
PW : None
URD: $.PROJECT.TEXT      Short SAVES OK
LIB: <-Normal->           Fixed *ENABLE
```

Accounts : 0->15 25 60->6F F0->FF

Expanded mode allows the details of each user to be edited. New users are initially created with no password set, boot option 0 (see under \*OPT4, Section 6.6), normal library and user root directories, and access to no accounts at all.

0 1 2 or 3 will set the boot option to that value. The boot option may also be set by the users themselves, unless the PW lock option has been set (see below).

- 8 will turn on and off an option to prevent users from saving a file shorter than 16 bytes in length with the SAVE or \*SAVE command. This option helps to avoid the problem where BASIC (for example) will save a null file if an attempt is made to save after pressing <Break> without typing "OLD".
- 9 will turn on and off an option to prevent users from changing their password and boot option. It could be useful to set this option for the default user, to prevent unauthorised users from changing the default password and option.
- S toggles system privilege off and on for that user. *Note that there must be at least one system privileged user on the system, or it will not be possible to change the password file thereafter.*
- D sets this user as the default user. There can be only one default user, so this command will change the default to this user. Keying D again will remove the default setting, so that there is no default user. Users logging-on to the File Server with unrecognised user identifiers will be logged on as the default user -- the system manager should set up a boot file to re-direct them, if necessary. If there is no default user, the error **User not known** will be displayed.
- E toggles an option requiring the user to type \*ENABLE before attempting \*DELETE with a wild-card specifier, as a safety measure.
- P prompts for a password. Users can also set up their own passwords with the \*PASS command unless the PW lock option has been set. Passwords can be up to 10 characters long, and have the same valid characters as file names.
- A prompts for account numbers. In response it is possible to specify a single account, several accounts separated by commas or spaces, or a range of accounts: for example 1-55 specifies all accounts from 1 to 55. If the line starts with a minus sign, the specified account(s) are removed from the user's list, otherwise they are added. Typing A <Return> will leave the user's accounts unchanged. Note that account 0 is a 'system' account as the root directory on each disc always has account 0: hence account 0 should normally *only* be allocated to the system manager.
- U prompts for a user root directory. This can be a path name going through several directories, and can be up to 80 characters long. Disc names can also be included to specify a disc; the default disc is the one in which the user is found in the password file. If <Return> is pressed, the <-normal-> option of a directory with the same name as the user identifier will be selected. If the specified URD is not found on logging-on, the user will be in the root directory of the default disc, even if another disc was specified. Wild cards can be used in a URD specification, although this is not recommended.
- L prompts for a default library directory for the user. This can be a path name going through several directories, and can be up to 80 characters long. If <Return> is pressed, the <-normal-> option of \$.LIBRARY on the lowest numbered disc drive is selected. If the specified library is not found, the default library will be set to the root directory on the lowest numbered disc. Wild cards can be used in a library specification, although this is not recommended. The character & can be used as a synonym for the matching URD if required.

To return to the list of users, press <Return>.

## Likely Errors:

### Key Locked

### Error 5 (05)

If the key switch on the front panel of the MDFS is not in the SYST position.

### Insufficient privilege

### Error 186 (BA)

If the user does not have system privilege.

## Compatibility Notes:

Not supported by Acorn systems.

## Syntax: CHAIN "EDITPRINT"

### Description:

The EDITPRINT program allows the system manager to set up details for the two printers which can be connected to the File Serve and to set the amount of system information.

Although anyone can run EDITPRINT (assuming that the system manager has set public read access to this file) to find the default settings, it is necessary to have system privilege, and the MDFS front-panel keyswitch must be in the "SYST" position to change the printer information.

This description merely describes how the program works : see Chapter 6 for advice on suitable values to set.

To adjust the printer settings, type:

**CHAIN "EDITPRINT"**

The program will respond with the main menu :

```

Edit logical printer details
Change system messages
Set up initial choices
Save changes and exit
    
```

Throughout the program, items can be selected from the menu by using the up and down cursor-keys to move the menu bar over an option. Pressing **Return** selects that item.

### Option 1 - Edit logical printer details

This will result in a list of *logical* printers being displayed on the screen, for example:

1.Microl	Parallel
2.Serial	Serial
3.Nobann	Parallel
4.	Serial
5.conden	Parallel
6.Epson	Serial
7.	Parallel
8.	Serial

The right hand column indicates which *physical* printer will be used: while the File Server only has connections for two printers, it is possible to have several printer names associated with each one.

When a printer is selected, its details will appear:

```

Name: MICROL
Printing enabled: Yes
Bannerfile: Banners.Parallel
Spool to Disc: Yes
Anonymous Users allowed: Yes
Account Ownership required: No
    
```

Again the menu bar highlights one item. Yes/No items can be changed by pressing space, while other items can be changed by typing a new value, followed by **return**. **Return** on its own writes any changes back to the File Server and returns to the main menu. Escape discards any changes which may have been made by

mistake.

**Name** is the name which users will quote to specify that particular logical printer. Printer names may be up to six characters long. The names PRINT, HOLD and AUTO are reserved and must not be given to a printer. If the printer name is blank (i.e. consists of spaces), that printer is disabled completely.

**Printing enabled** controls whether output sent to this particular printer will be printed. It does not prevent users from generating output, which will be spooled to disc.

**Banner file** gives the name of a text file which controls the *banner* that is printed at the top of all printer output. The various possibilities for the contents of the banner file are described in section 6.3.2. The file name is looked up starting from \$ on the first disc drive, so **banners.fancy** would be equivalent to **\$\*.banners.fancy**.

**Spool to Disc** controls whether printing starts as soon as some data arrives, or whether it is spooled onto disc and only printed when the whole document has arrived.

**Anonymous users allowed** control whether users who have selected this logical printer, but are not logged on to the File Server, may print.

**Account ownership required** controls whether a user requires a specific account number to select this logical printer; beware that if this printer is listed under initial choices then the account ownership check will be bypassed.

## Option 2 - Change System Messages

This enables you to set the level of system messages.

```
0 = serial
1 = parallel
```

```
System message    Parallel
Message Level is 0
```

N.B. System error messages are ALWAYS sent to the printer.

By typing zero or one, the printer port used for system messages can be selected. It is not possible to disable system messages altogether, as the system has to have some way of displaying warnings of impending failures.

If the menu bar is moved down, a list of the possible message levels appears. The selected level can be changed by typing a number followed by **return**.

```
0 = off
5 = logon/logoff
7 = errors
10 = maximum users & *commands
11 = load/save
15 = *cat and opens
128 = aborted loads
130 = Fn codes
150 = net errors
200 = disc read/write
250 = all sucessful net transactions
255 = all activity to JPROC
```

```
System message    parallel
message level is 0
```

These represent cumulative levels of system messages (7 includes the message of 5 and 0). Although the system message level may be set to 0, system messages after catastrophic errors will still appear on the printer.

The usual message level is zero.

### Option 3 - Set Up Initial Choices

This option allows you to specify the default printer for users who do not select a particular printer, and to indicate the first and second choices of printer when **AUTO** is selected as a printer.

The screen will display:

Priority of Printer	No	Printer
	0	STOP
1st Microl	1	MICROL
2nd Epson	2	SERIAL
	3	Nobann
	4	
	5	conden
	6	Epson
	7	
	8	
Default AUTO	9	HOLD

New default choice.  
Press 0-9 to select printers

The display on the right hand side of the screen lists the available printers. If the second choice is set to zero, then **AUTO** will be equivalent to the first choice printer. If the first choice is set to zero, the **AUTO** printer will be disabled completely.

Remember that any user can select and print to the **AUTO** printer, bypassing any restrictions that may be placed on the first or second choice printers. Hence it is not usually sensible to specify as first or second choice a printer which has account ownership required.

The **default** printer is the one which will be used if a user sends data for printing without selecting a particular printer. The File Server still checks whether the user is permitted to use that printer, so restricted access on the default printer will prevent some users from printing without explicitly selecting a printer to which they do have access.

### Option 4 - Save Changes And Exit

This option puts into effect any changes which have been made through the other **Editprint** options. Note that the changes have already been written to disc, so leaving **Editprint** without using this option will not discard the changes: they will come into effect next time the fileserver is re-started.

### Likely Errors:

#### Key Locked

#### Error 5 (05)

If the key switch on the front panel of the MDFS is not in the **SYST** position.

#### Insufficient privilege

#### Error 186 (BA)

If the user does not have system privilege.

### Compatibility Notes:

Not supported by Acorn systems.

## **Syntax: \*FAST**

### **Description:**

This program turns the BBC Microcomputer into a terminal to any suitably equipped host computer: communication is through the Econet network. The program is supplied as standard on disc (in the library directory), but a ROM version is also available for the convenience of Hard Disc users in the event of a corrupt hard disc. The ROM version should be fitted to a BBC micro as a standard *sideways ROM*.

The most common use of \*FAST is to communicate with a SJ Research File Server system, when the latter is in Utility Mode.

The FAST program will prompt:

```
Station number to attach to :
```

Type in the station number of the File Server or other host computer (usually 254).

Both versions of FAST operate in the same way, but when using the disc version with the filesaver in utility mode, it is important to load the program (by typing **\*FAST**) *before* the filesaver is placed in utility mode. The usual way to do this is to type **\*FAST**, ensure that the key switch is set to the *system* position, and then type **\*FINISH**. After a few seconds, the filesaver will light the utility mode lamp, and the station number can be typed to complete the connection.

The EPROM version will re-start if <Break> is pressed: it will be necessary to press <Ctrl-Break>, or to type **\*BASIC** after the **Station number** prompt, to exit from the program.

The RAM version will stop if <Break> is pressed. It may be re-started by typing **CALL &2800**.

In either version, \* commands may be entered at any time, preceding them by <Shift-f1>, or by typing **\*<command>** after the **Station number** prompt.

### **Example:**

```
*FAST
```

```
Station number to attach to : *FINISH  
Station number to attach to : 254
```

followed by output (if any) from station 254.

### **Likely Errors:**

Normal BBC Microcomputer errors will be preceded by **OS Error**.

No response, or the error **Not listening**, is usually caused by the File Server not being in Utility Mode, or by being attached to the wrong station.

### **Compatibility Notes:**

Not supported by Acorn systems.



---

## Syntax: CHAIN "FORCER"

### Description:

This program sends a string to the keyboard input of one or more stations -- i.e. it will force them to execute any command sent. It could also be used to send a message (preceded by \*|| which causes the O.S. to ignore the rest of the line).

It is possible to send the command to all stations of a group. Load FORCER (it is in the release library), and re-type line 50 to contain all the stations which are to be included in the group:

```
50 DATA <station number>,<station number>, ... END
```

When the program is run, it will prompt:

```
Station (<RETURN> for all) :  
then,  
Command :
```

If <Return> is typed after the first prompt, then the command will be sent in turn to all the stations in the DATA statement.

This program is primarily intended as a basis for developing more sophisticated programs, but it may be useful in its present form.

### Example:

```
CHAIN "FORCER"  
Station (<RETURN> for all) :  
Command : LOAD "EXAMPLE1"  
  
Sending to station 1  
Sending to station 3  
.  
.  
.
```

It is wise to hide away the \*PROT utility that sets the protection byte to stop direct operations, otherwise users can prevent FORCER from working on them.

### Likely Errors:

The program uses the same Osword call as \*NOTIFY, and can give rise to the same errors, e.g. Not listening if one of the stations is absent, switched off or protected.

### Compatibility Notes:

Supported by Acorn systems.

---

**Syntax: CHAIN"LISTUSERS"****Action with Wild Cards:**

Not applicable

**Description:**

This program lists all the users, held in the %PASSWORDS file, with their accounts and any special options that are set. This is especially useful for providing the system manager with a list of information so that he knows which accounts are spare to allocate.

To send the output to the printer type CTRL B before running the program.

**Example:**

**CHAIN"LISTUSERS"**

User name	Boot opt. Default settings	Accounts
ANDY	exec No short SAVES	00,06,20,2F,30-3F,50
ARG	Permanent *ENABLE	00-FF
SYST	** System privilege **	00-FF
TONY	Permanent *ENABLE URD=\$.RELEASE.FS*.ISS023	20-FF

**Likely Errors:****Key Locked                      Error 5 (05)**

If the key switch on the front panel of the MDFS is not in the SYST position.

**Insufficient privilege              Error 186 (BA)**

If the user does not have system privilege.

**Compatibility Notes:**

Not supported by Acorn systems.

---

**Syntax: \*LOGOFF****\*LOGOFF** <username> | <station>{,<username> | <station>}**Action with Wild Cards:**

Will apply the wild-card to user names, to log off all occurrences of all matching user names.

**Description:**

Used without a parameter, this command has exactly the same effect as **\*BYE** except it will terminate any printer job currently being generated from that station.

When followed by one or more user names or station numbers (separated by commas), any user logged on using one of those user names or at one of those stations (except for the station originating this command, if that user is included in the list) will be logged off.

Wild cards are permitted, so that:

**\*LOGOFF #IM**

will log off all occurrences of users JIM, KIM and TIM etc.

**\*LOGOFF \***

will log off everyone except the person typing the command.

Anyone may use **\*LOGOFF** without a parameter, otherwise system privilege is required.

**Likely Errors:****Key Locked****Error 5 (05)**

If the key switch on the front panel of the MDFS is not in the SYST position.

**Insufficient privilege****Error 186 (BA)**

If the user does not have system privilege.

**Compatibility Notes:**

Not supported by Acorn systems.

---

**Syntax:** \*PGO | \*PGO <logical printer name>

## Description:

This command starts the printer server printing after a \*PSTOP command has been given. The print job will be restarted from the beginning. The command is especially useful if the printer runs out of paper whilst printing a job.

System privilege is not required, nor does the key need to be in the SYST position, however access to either the main or auxiliary account of the %PRINTQ directory is required.

## Example:

\*PGO PARALL

## Likely Errors:

**Insufficient access**                      **Error 189 (BD)**

If the user does not have access to either the main or auxiliary account of the print queue directory.

## Compatibility Notes:

Not supported by Acorn systems.

---

**Syntax:** \*PSTOP | \*PSTOP <logical printer name>

## **Description:**

This command stops output on the specified printer, or all printers if no name is given. It is intended for use where the printer has jammed or is producing unwanted output. Printing is suspended, and any file which was printing is closed. The printer manager can then inspect the contents of the print queue, delete unwanted jobs, change the order of priority of jobs, and then use \*PGO to resume printing.

Note that printers have an internal buffer which will not be cleared, so the output will continue for a short period after the command has been given. Most printers have a facility for setting the size of their internal buffers, this should be set to a minimum.

## **Example:**

**\*PSTOP**

## **Likely Errors:**

**Insufficient access**

**Error 189 (BD)**

If the user does not have access to either the main or auxiliary account of the print queue directory.

## **Compatibility Notes:**

Not supported by Acorn systems.

---

## Syntax: CHAIN "SETTIME"

## Description:

This program sets the internal real-time clock in the File Server.

The program will prompt for the current date and time, and will then ask the user to press the space bar to set the clock to the time entered, allowing accurate synchronisation with the speaking clock or Greenwich pips.

## Example:

The program will prompt as follows:

```
This program sets the File Server clock
It may only be used by Privileged Users
```

```
Use *TIME to see the present setting
```

```
Year : 1984
```

```
Month (1=Jan,12=Dec) :9
```

```
Day (1-30) :9
```

```
Hours (0-23) :9
```

```
Minutes (0-59) :18
```

```
Ready to set time to:
9/9/84 9:18
```

```
Press SPACE to set time, ESCAPE to abort
```

To set the time again with the same value, type **GOTO 370** after running the program (but note that this line number may change in different versions of SETTIME).

## Likely Errors:

### Key Locked

### Error 5 (05)

If the key switch on the front panel of the MDFS is not in the SYST position.

### Insufficient privilege

### Error 186 (BA)

If the user does not have system privilege.

## Compatibility Notes:

Supported by Acorn Level 3 systems.

**Syntax: CHAIN "SIZER"****Description:**

This BASIC program searches through a directory tree, finding every file and sub-directory. It will print out every file with its size in units of 1 kilobyte, and then give a grand total.

Anyone can use SIZER within their own directories, but system privilege is required if the root directory \$ is specified for the directory to search, as SIZER will try to inspect the password file.

SIZER prints the true size of each file, which is the amount of disc space occupied by the file. This is not necessarily the same as the file's logical length, or *extent* (EXT# in BASIC). The disc space is used up in units of 1K, and so a file with an extent of 1 byte still occupies 1K on disc. In addition, large files require some extra space for the system to hold pointers; again this is included in the true file size.

In some cases though the size of the file may be *less* than the extent. This is because a file may be opened, some information written, and then the pointer moved to leave a large gap before further writing. Such sparse files will be allocated space in 1K blocks only where data has been written, and the size of the file as found by SIZER will be only that of the blocks actually allocated.

**Example:**

```
CHAIN"SIZER"
```

```
Directory to find size of?$
```

```
Directory $                4k
Directory ACCOUNTS        1k
  Ashton                   1k
```

```
.
.
.
.
.
```

```
Total                    568k
```

**Likely Errors:****Insufficient access****Error 189 (BD)**

If there are files or sub-directories without public access R, and the user of SIZER does not have access to the account number of these.

**xxxx not found****Error 214 (D6)**

If there are files or sub-directories with access letter P, and the user of SIZER does not have access to the account number of these.

**Key Locked****Error 5 (05)**

If the key switch on the front panel of the MDFS is not in the SYST position and \$ was specified.

**Insufficient privilege****Error 186 (BA)**

If the user does not have system privilege and \$ was specified.

**Compatibility Notes:**

Not supported by Acorn systems.

## 4.4 The File Server program

The File Server Program is stored in a file on a disc called **\$.FS**. The file may or may not be present on any of the discs that you use, floppy or hard. It is a large file, about 100kilobytes long, containing both the Utility Mode program and the File Server program, loaded by the system after the MDFS is switched on. It does not matter what its access string or account numbers are, as the system will always read it. It has a Checksum appended to it so that the system will not try to execute code which has been corrupted (e.g. by someone modifying the file). Being an ordinary file, it can be deleted (to leave more space on a floppy disc) or copied from another disc using **COPIER**. Remember to keep at least one disc containing **\$.FS**, to use when switching on the system or entering utility mode.

A situation you must guard against is where you have several different versions of the File Server code on different discs. Depending which disc the File Server is booted from, the version of code will be different. Old versions of the code may contain bugs which later versions have solved, and you can get very confused when the old bug reappears even when you are apparently running the new software. The solution is to delete all old versions of the file **\$.FS** when an updated version is recieved.

Since the file contains the Utility Mode software and the File Server software, there are in fact two version numbers. To get the version number of the File Server code *currently executing* type **\*VERS**. The version number of the Utility Mode is printed at the top of the Main Menu (which you see when you attach with **\*FAST**).

The following examples assume that the disc called **MASTER** is the master floppy disc sent with the MDFS, or a later upgrade, and **HARD1** is a winchester disc.

### 4.4.1 Copying new File Server software onto a winchester disc

You will have received a floppy disc in MDFS format, the master disc. Connect a floppy drive to the MDFS and put the disc in, making sure that there are no other floppy discs in any other drives. Start the File Server, log-on as a system-privileged user, and type:

<b>*DIR \$HARD1&lt;return&gt;</b>	selects the winchester
<b>*ACCESS FS&lt;return&gt;</b>	unlocks the file, if necessary
<b>*DELETE FS&lt;return&gt;</b>	

(We need to delete the file before running **COPIER** because of file access problems)

<b>CHAIN"COPIER"&lt;return&gt;</b>	
Source Filing System: <b>*&lt;return&gt;</b>	
Dest. Filing System: <b>*&lt;return&gt;</b>	
File Name: <b>\$MASTER.FS&lt;return&gt;</b>	read the fileserver code from the floppy
New Name: <b>\$HARD1.FS&lt;return&gt;</b>	and write it to the winchester
File Name: <b>&lt;escape&gt;</b>	

<b>*ACCESS FS PL&lt;return&gt;</b>	Remove access to the file
------------------------------------	---------------------------

You will now be able to start the system with the new software without using the master disc.

# Chapter 5: Setting up users

---

## 5.1 Overview

Your system will have a number of different users to whom you will want to be able to give facilities to create files for themselves, to read certain communal files (for example library programs) and to have selective access to other users' files.

The list of authorised users in a SJ Research File Server is kept in a file called the *password file*. This file can be read and saved only by someone with *system privilege*; normally only the system manager himself and only when the front panel key-switch is turned to the SYST position. The password file contains information about each user: his password, any accounts he has access to, and administrative information concerning start-up (boot) options, library directories and user root directories.

If someone logs-on to the system, and his name does not appear in the password file, then he will be logged on as the *default user*, if one has been set up by the system manager using EDITPASS (see Section 5.3). If no default user has been set by the system manager, the user will receive the message **User not known**.

When a user listed in the password file logs-on, any password he quotes will be checked against the one in the password file, before the log-on is allowed to proceed. He will then be given any rights and privileges listed against his name in the password file. The system will then search the disc on which the user's password file entry was found, for the *User Root Directory* specified for that user in the password file, which by default has *that user's name*, and will set this to be the currently selected directory (see Section 3.3 under \*I AM for details). If no appropriately named directory is found, the disc root directory will be selected.

As described more fully in Section 3.3 (under \*ACCESS and \*ACCOUNT), the *account(s)* to which a user is given access, control two things:

First, every file (or directory) has an account number, and *if a user has access to this account, then he is an owner of that file (or directory)*. Only an owner may create files in a directory, and only an owner may delete a file or change its access letters (see Section 3.3 under \*ACCESS command). Note that there can be *more than one owner* of a file (or directory), simply by allocating access to its account to more than one user - this can be useful for communal files in a project.

Second, each account has a *credit balance* of storage space, and an attempt to create a file which would cause that balance to become less than zero will be prevented, and cause the error **Account bankrupt**.

### 5.1.1 Keeping a List of Users

It is wise to plan your list of users, and the accounts for them, on paper and keep it up to date. There is no security required for account numbers and users' names, and even a moderately sized system can have more users and accounts than can be displayed on a screen.

User names may have up to ten characters, which may include letters, numbers and dashes, and must start with a letter. Normally the user's name would be his own surname or initials. However, user names must be unique in the system, so you may wish to add figures to the end of a name.

Account numbers range between 0 and 3FF (hexadecimal), but you may of course ignore the hexadecimal part and just use numbers up to 399. Allocating account number 0 gives ownership of the system root directory, so *account 0 should be allocated only to system privileged users*.

### 5.1.2 Entering Users on to the System

Switch on your File Server unit and log-on as a system privileged user. If you have just taken delivery of the system, use the name SYST, with the password SYST. In other words, at a BBC Microcomputer on the network, type:

```
*I AM SYST SYST
```

You will then need to edit the password file to enter your list of users and their accounts, then create *user root directories* for each of them, and then set the *account number* of each user root directory to the same as that allocated to the user, so that the user has owner access to this directory. These three operations are described below.

## 5.2 Listusers

The system manager should keep a list of all the users and their accounts as they are created. When the system manager loses track of the account structure then **Listusers** can be run to display all the users and their account numbers. As with Editpass, this program manipulates the password file so the usual precaution of \*PROT and turning off the BBC microcomputer, after running the program, should be observed.

To send the output to the printer press **CTRL B** before running the program. For more information see the system managers utility programs, section 4.3.

## 5.3 Editpass

To edit the password the key must be in the SYST position, you must have system privilege and have access to account 0. The Editpass program is also detailed in section 4.3.

### 5.3.1 Using Editpass

Please see Section 5.3.3 if you are likely to require password files on more than one disc (for example - if you wish some users to have access to the system only when a particular floppy disc is present).

A BASIC program called EDITPASS (fully described in Section 4.3 under EDITPASS) is provided in the library of the system: to use it, check that the front panel key-switch is in the SYST position and type:

```
*PROT      (PROT prevents other users peeking your machine)
CHAIN "EDITPASS"
```

The program will ask whether you want to edit the password file from disc (option E), or to start a new password file (option N). As delivered, the system disc contains a password file with at least the three users SYST, FRED and BOOT, so you would normally edit this one - it is usually preferable to edit the existing file (unless you have had to delete the password file, in which case use N).

The contents of the existing file will now appear, along with a menu of commands. Use command A to add new users. To save time, enter all the foreseeable users this stage, ending each one by typing <Return>, and end the list by typing <Return> on its own.

To allocate accounts, select users one by one, using the brown cursor control keys to move the list past the cursor. After selecting a user, type X to enter the *expanded mode* of EDITPASS. The screen will now display the accounts allocated to that user, and will also display other information relating to that user. Press key A to enter the account(s) to be allocated to the user. A list separated by spaces, or a series with a dash between numbers, may be entered. To remove accounts, prefix the number(s) with a dash (as minus sign). Press <Return> on its own to return to the main list of users, and repeat the account setting for other users.

Before finishing with EDITPASS, check these points:

- a) There must be at least one system privileged user on at least one disc (see Section 5.3.3), otherwise it will not be possible to read or save the password file in the future. If you do not wish to use the name SYST, then enter a suitable name for yourself. Type X, give yourself system privilege (type S), and allocate yourself all accounts (type A 0-3FF). Give yourself a sensible password that no-one is likely to guess.  
*PHILIP JANE,  
WARR, N.*
- b) There must be a user on at least one disc (see Section 5.3.3) with access to account 0, otherwise it will not be possible to create user directories or edit the password file. Ensure that you have given yourself (and any other system privileged users) access to account 0. If you omit to do so, you may have to delete the password file and start again - a tiresome procedure.

The EDITPASS program will display a warning if either (a) or (b) above are not complied with. Note however that you do not need a system user on every disc - in fact it can be a useful security aid to lock away floppy disc(s) with a system user when one is not required. See Section 5.3.3.

- c) It is wise for the system manager to have a different name for day to day use from the one he uses for system privileged operations. This means that someone watching him log on will probably not see the important password.
- d) If you are going to use the name SYST for the system privileged user, then change the password, otherwise a casual reader of this manual will be able to log on with system privilege.
- e) If you want to have a *default user* on the system, set this up with option D in the expanded mode for the appropriate user. If there is no default user, unrecognised user identifiers will get error **User not known**; if there is, they will be logged on as the default. It is normally recommended that you do not allocate any accounts to the default user, so that someone logging on as an unrecognised user will not have owner access to anything.

- f) If you want anonymous printing enabled then there must be a user ANONPRINT or a default user. The corresponding logical printer should have the **anonymous printing allowed** option set to yes (see section 4.3 under EDITPRINT and section 6.3).

When you have completed the editing of the password file, return to the list of users, and press S to re-save the file. The program will display the name of the default user (if any) and a warning if there exists no user with system privilege or access to account 0, then will prompt Save? (Y/N) - check that the user list is correct, and that the default user and system privileged user(s) are correct before pressing Y.

*After using EDITPASS, type \*BYE and switch off the BBC Microcomputer. The password file remains in memory after EDITPASS, and could be looked at by an unauthorised user otherwise.*

### 5.3.2 Keeping a Security Copy of the Password File

It is wise to keep a copy of the password file elsewhere, so that the main file can be restored quickly if an accident does happen. An easy way is to use the BASIC program COPIER (described fully in Section 3.3), to copy the file either to a floppy disc on a suitably equipped BBC Microcomputer, or to a file in the File Server. In the latter case, set the access status (see \*ACCESS in Section 3.3) of the copy to PWR/, and give it an account number that *no-one else has access to* (e.g. Account 0).

The file itself is called %PASSWORDS, and is not in any particular directory. If a mishap does occur, use COPIER again, copying from the backup copy, back to %PASSWORDS.

After using COPIER for this purpose, type \*BYE, and switch off your computer, since the file will be left in the computer's memory.

### 5.3.3 Password Files on more than one Disc

On a Modular Disc File Server, the system will search through the password file on every disc, starting with the hard disc drives (if fitted) and then the floppy drives, until a match is found with the user name. It is possible to place users on specific floppy discs (using the \*SDISC command before or during running EDITPASS), so that they cannot log on unless that disc is in the File Server. In addition, it may be desirable to place the system privileged user(s) and/or all users with access to Account 0, on one particular floppy disc - which can be physically locked away when system privileged operations are not required.

When a user logs on, the system will search for his name in the password file on each disc, starting from the first disc alphabetically. When it finds this user identifier, it will search the same disc (and no others) for a user root directory. Hence a user's entry in the password file and his root directory should appear on the same disc, otherwise this automatic selection will not take place.

See Section 3.3 under \*I AM command for full details of the action of logging on. Note in particular that it is not desirable for a particular user to appear in the password file of more than one disc, since the stored password and account information may be different on the various discs, and the version used by the system will be the one in the lowest numbered drive - this could change at random for the user depending on the drives in which floppy discs had been inserted, and information on the second hard disc would never be used.

If any disc in the system does not have a password file on it, the File Server will give system privilege to anyone attempting to log on. The disc formatter in Utility Mode will automatically create a null password file on every disc, so that this cannot happen accidentally. If the password file has to be deleted (this is done from Utility Mode - see Section 7.3.8), ensure that the system is left with at least a null password file %PASSWORDS on every disc (with, of course, a 'real' password file on one or more discs). The null file can be saved with the program EDITPASS - after selecting the appropriate disc. Log on as a system privileged user (if one exists in any password file at this stage) with the front panel key-switch in the SYST position, type CHAIN "EDITPASS", then N for a new file, then S to save an empty file.

## 5.4 Finishing the process

Now that the password file has been defined it is necessary to create the directory structure and then set the account numbers to coincide with the *user root directories*.

### 5.4.1 Creating User Root Directory

You need to have access to account 0 to create directories in the root, as described in Section 5.3.1 above. In addition, if you are going to allocate accounts to users, you will need access to all other accounts. Check that you are in the disc root directory (that is the directory in which all the user root directories appear) on the desired disc, by typing:

```
*DIR $ or
*DIR $<disc name>
```

Then create a directory for each user, using the CDIR command:

```
*CDIR ALLEN
*CDIR BURTON
```

and so on. At the end, check that the directories exist by typing \*CAT, which will give a list on the screen of all the directories in the root.

### 5.4.2 Setting the Account Numbers of User Root Directories

You have already allocated one or more accounts to each user whilst editing the password file. In order for the users to have owner access to their own user root directories, you need to give the corresponding account number to their root directory. The File Server will then automatically take care of accounting, by the rule that any file (or sub-directory) created will have the same account number as that of the directory that it is in. A user may change the account number of a file, but only if he has access to another account to change it to.

Before setting account numbers, ensure that each account that you are going to allocate has some credit. See Section 3.3 for information about \*STATEMENT and Section 4.3 on \*CREDIT.

The root directory \$ on each disc always has account number 0. By the above rule, all the directories will initially have account number 0, and hence only users having access to account 0 will own them, allowing them to create files in them.

To change the account numbers of these directories, use the \*ACCOUNT command. Suppose that you had allocated account 22 to user ALLEN, and account 23 to user BURTON. Then type:

```
*ACCOUNT ALLEN 22
*ACCOUNT BURTON 23
```

and so on. You need to have access to all accounts to be able to do these allocations - how to do this is explained in Section 5.3.1, particularly in paragraph (a).

You may get the error message **Account Bankrupt** whilst running the \*ACCOUNT command. It will be necessary to credit the account using the \*CREDIT command (see below).

If you have set a default user in EDITPASS (sections 5.3.1 & 4.3), you will probably want to set up a user root directory for the default, and put a !BOOT file in it (see Section 3.3 under \*OPT4 for details). In this case, it is recommended that you leave this directory with account number 0, so that only you can change it.

Alternatively, you may wish to allocate some 'scratch-pad' filing space to anonymous users, in which case both the default user and the corresponding user root directory should be given an account (with the balance suitably set using \*CREDIT) to permit this.

### 5.4.3 Distinction between Users' Names and User Root Directories

There is the possibility of confusion between a user called FRED (for example) and a user root directory (or URD) called FRED. The concept of a user exists *only in the password file*, and is the means of identification for the purpose of allocating accounts, privileges and other options.

At log-on, the system will search the disc on which the user was found in the password file for a URD, as a convenience to that user and others. (Note that this is different from Acorn systems, which search *all* discs for a URD). This root directory will have the same name as that selected by the URD option in the password file. If no particular selection has been made, the "normal" entry in the password gives a URD with the same name as the user: for example, user MARY is assumed to have URD MARY unless another URD name has been entered into the password file. Others can gain public access to MARY's files (assuming that she has allowed public access) with commands like

```
LOAD "$ .MARY .PROGRAMS .EXAMPLE1"
```

Note however that there is actually no need to allocate a separate URD to every user. The allocation of accounts is the *only* thing that determines whether a user has owner access to a file.

For example, for a sixth form project with three people working on it, it may be useful to have a root directory called PROJECT. Suppose the three users had user names JOHN, KATHY and MARY, you might allocate them all account 98. Then set the account number of PROJECT to 98, and all three users have owner access to directory PROJECT, and can all create files in it.

At the same time, any or all those users can be given owner access to other files or directories, by allocating them other accounts; so that you could also allocate account 43 to user MARY, and then set up a directory called MARY, with account number 43 - only MARY would have owner access to this latter directory. The URD option in the password file is used to specify which of the directories that MARY has owner access to is her URD.

If there is no directory on the disc to match the URD specified in the password file for a particular user, say JOHN, then when JOHN logs on, the File Server will select the root directory \$. Note that only the first disc on which the user was found in the password file will be searched. JOHN will need to select directory PROJECT explicitly before he can begin work, by typing:

```
*DIR $.PROJECT
```

Alternatively, the system manager can set the user root directories of JOHN, MARY and KATHY to be \$.PROJECT by pressing U at the expanded information stage of Editpass.

### 5.4.4 Crediting or Debiting an Account

When the system manager formats a new disc, the formatting program will initially set account 0 with credit equal to the disc size, and all the other accounts with zero credit. See Chapter 7 for a full description of the Utility Mode commands and how to use them. To find out what the credit balances are, log-on as a user with access to all accounts, and type:

```
*STATEMENT
```

Disc	0
Account	Balance
50	5K

Disc	1
Account	Balance
50	10100K

This will produce for each disc a list of all the accounts to which a user has access, with the associated credit balances. For the system manager it would be wise to send this information to a printer, so type:

**\*PS**                    this selects the network printer  
**<Ctrl-B>\*STATEMENT**   **<Ctrl-B>** means press B while holding down the CTRL key  
**<Ctrl-C>**

The network printer will then produce a copy of the screen. To change the balance of any account, use the DEBIT and CREDIT commands. For example:

**\*CREDIT 43 500**

will add another 500 kilobytes of space for use by files with account number 43. Similarly **\*DEBIT** will subtract space from that allocated. These two commands require system privilege and the front panel key-switch must be in the SYST position.

These commands only change the outstanding credit balance of available space. Since the credit balance cannot exceed 65535 kilobytes, the command:

**\*DEBIT 44 65535**

is guaranteed to wipe out any credit balance left to account 44.

When a file is created, the account corresponding to its account number is debited by an amount equal to the size of the file (but not its extent - see Section 3.3 under OPENOUT). If there is insufficient balance to allow saving of a file, the error message **Account bankrupt** will be produced, and the user must either delete something, or move some files to another account - assuming that he has access to another. No account can be debited below zero credit, so deleting files will always give some positive balance.

On a system which has more than one disc in use, the STATEMENT utility will print a list of the accounts to which the user has access, separately for each disc. The user can however only save files on discs which contain one or more directories with account number(s) to which he has access.



## 5.5 Password File Management System

### 5.5.1 Overview

#### Batch mode Editor Documentation:-

- 5.5.1.1 Memory Requirements
- 5.5.2 \*CONVERT
- 5.5.3 \*MERGE
- 5.5.3.1 !mkdir and !rmdir
- 5.5.4 \*GENERATE
- 5.5.5 Keywords
- 5.5.6 Mod-file Examples
- 5.5.7 Errors
- 5.5.8 Formal File Definition
- 5.5.9 Known Problems

### 5.5.1 Overview

The password file management system software consists of the following programs, all of which are found in the directory \$ . SYSPROGS of the release disc:-

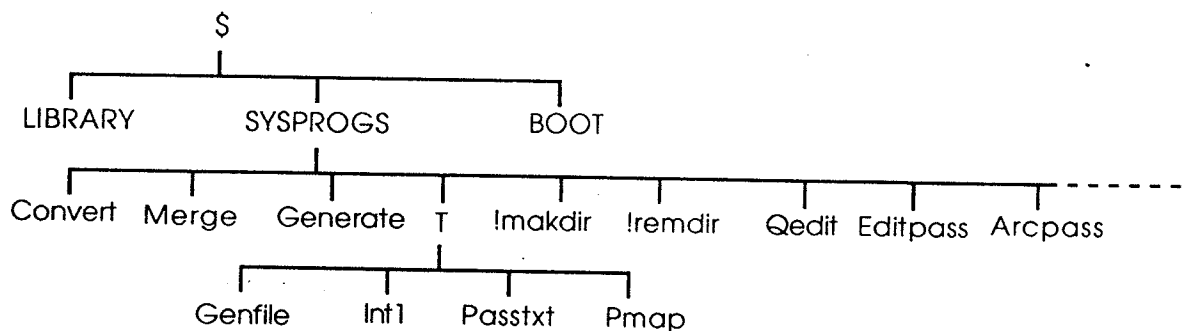
#### a) The batch mode editing suite:

CONVERT (Machine code program)  
MERGE (Machine code program)  
GENERATE (Machine code program)

#### b) The interactive editors:

QEDIT (BASIC program)  
EDITPASS (BASIC program)  
ARCPASS (Archimedes BASIC program)

The directory stucture is shown thus:-



The existing EDITPASS program restricts the size of the password file to the size of the memory in the local computer, and this typically allows around 200 users. There is now a version of version of this program (called ARCPASS) for an Archimedes allowing about 7000 users. The batch mode editor and QEDIT are a means of editing large password files on standard BBC microcomputers.

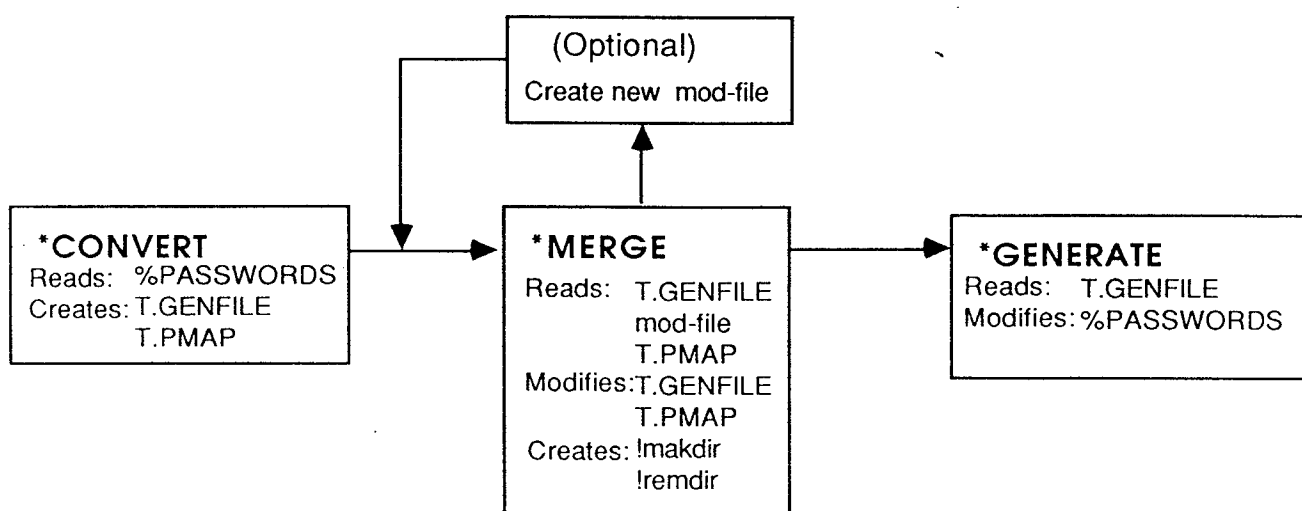
QEDIT is a version of EDITPASS which allows the password file to be edited on a user-by-user basis. The password file is not held in the local computer; individual user entries are modified and then written back to the password file directly, so the restriction on file size is removed. However, QEDIT does not allow you to insert or remove users, or to change the URD or LIB strings.

With the batch mode editor, the system manager prepares a text file (the *mod-file*) containing instructions for modifying the password file. The commands available can be very powerful; for instance, the system can automatically allocate a spare account number, create the appropriate user directory, set its account number

and credit that account. The same process can be repeated automatically, so with little more than a list of names, an entire class can be entered onto the system in a matter of minutes.

The batch mode editor uses a three-stage process: \***CONVERT** converts the (machine-readable) password file %**PASSWORDS** into a (human-readable) text file. \***MERGE** combines this with the mod-file to produce another text file. \***GENERATE** converts this text file back into a machine-readable format file, which it then installs on the appropriate disc.

The process is shown in the diagram below:-



### Important

Since all of the passwords are stored in the text files, it is very important that only the system manager has access to them, and they should be treated with as much respect as %**PASSWORDS** itself. Each of the programs protects the machine from remote network operations to stop unauthorised people being able to read the files, but security is only as good as the system manager makes it. The T directory should be set to Private (\***ACCESS** T +P). **Only %**PASSWORDS** is protected by the key: the other files are only protected by the main file access controls.**

### Off-line / Off-site Operation

An advantage of the batch mode editor is that it can be run off-site using a local disc filing system (DFS or ADFS), thus reducing the risks of security breaches. \***CONVERT** is run (on the network), T.GENFILE and T.PMAP are copied onto local disc: the network copies should then be deleted. All the edits (i.e. preparing mod-files and running \***MERGE**) can then be done whilst the computer is disconnected from the network. T.GENFILE, !mkdir and !rmdir are copied back onto the fileserver, and \***GENERATE** is run. T.GENFILE should then be deleted from the fileserver.

### General Suggestions

If the password file is fairly small then EDITPASS can be used. If an Archimedes is available then ARCPASS can be used (on virtually all sizes of password file). If the file is too big for EDITPASS then QEDIT can be used, subject to the limitations of QEDIT itself.

If a large number of users are being added or modified, then, whether the password file is large or small, we recommend that you use the batch mode editor. For extra security the batch mode system should be used off-site.

## Temporary files created by the batch mode editor

There are a number of temporary files used by the batch mode editor which are all held in the directory T. T.GENFILE and T.INT1 should be deleted (for security reasons) after a session has finished. The files are:-

T.GENFILE    T.PMAP    T.INT1    T.PASSTXT

T.INT1 and T.PASSTXT are temporary files created and used only by \*MERGE. The latter is the updated version of T.GENFILE and is normally \*RENAMED as this before MERGE exits. However, if MERGE fails it is possible that both T.GENFILE and T.PASSTXT will remain. Thus T.PASSTXT may be deleted at any time (except while MERGE is actually running).

There also are two files created by \*MERGE that will require be to \*EXECed by the user, which are:-

!mkdir        !remdir

### 5.5.1.1 Memory Requirements

\*MERGE requires HIMEM at &7C00 or greater. On a BBC microcomputer without shadow RAM, MODE 7 is required (and will automatically be selected if this is not already the case). If HIMEM is less than &7C00 and you have shadow RAM, MODE 131 will automatically be selected.

N.B. If HIMEM is less than &7C00 and you load \*MERGE, you will see the program being loaded into the screen. Normally, this does not matter because the first thing that the program does is to change to a different mode. However, if you in addition have \*OPT 1 1 set, there is a fair chance that the text printed by the OS will actually overwrite the loaded program, which will then crash.

You are therefore warned against using \*OPT 1,1.

If you are using a RISC OS computer then the \*CONVERT, \*MERGE, \*GENERATE suite of programs can be run using the 65Tube module.

### 5.5.2 \*CONVERT: Converting the existing password file

Syntax: \*CONVERT [<Disc name>]  
System Privilege Required.

Using the program CONVERT, a (human-readable) text file T.GENFILE is created from the current password file, %PASSWORDS. The discname is recorded in this file. In addition the program will create the file T.PMAP which contains a *bit-map* of all the personal account numbers currently allocated in the password file.

Note that users are allowed to modify some aspects of the password file themselves (by using \*PASS or \*OPT 4,n), so you should not use the old copy of T.GENFILE but create an up-to-date copy (you *cannot* use the 'last update date/time' to see whether the file has been directly modified in this way). However, if you wish to use a sequence of \*MERGE operations you must only run CONVERT once (if you run CONVERT immediately after MERGE, the file T.GENFILE will be overwritten and any modifications will be lost).

Having typed \*CONVERT the software will respond with :-

```
*Convert
Version 1.12, (C) SJ Research
Converts %PASSWORDS into text file T.GenFile
also makes T.pmap
```

For every ten users processed a dot will be printed.

If you want to look at the resulting file, type `*TYPE T.GENFILE`. To convert a password file other than the one on your currently selected disc specify the disc name after the `*CONVERT` command.

e.g. `*CONVERT MAIN`

### Corrupt %PASSWORDS files

CONVERT will give a warning when it finds corrupt URD/LIB pointers (i.e. those that point off the end of the file; pointers that point to other random places in the file could produce warnings of the URD/LIB text exceeding 80 chars).

## 5.5.3 \*MERGE and the mod-file

Syntax: `*MERGE [<mod-file name>]`  
System Privilege Not Required.

`<mod-file name>`, if not specified, defaults to `MODE`.

Changes to the password file are made by creating a *mod-file*. This file should contain a *mode* keyword, telling MERGE whether to add new users, modify existing users or remove existing users. There is then a section defining attributes that should apply to the new users: these are called *global* assignments. When removing users, this section is obviously not required. Then follows the list of usernames on which we wish to act. Each username is followed by a section (enclosed in curly brackets { } ) that defines actions to be done to that user only (these are called *local* assignments).

### Creating a Mod-file

A standard ASCII text editor is required. We suggest using Acorn EDIT (supplied with a BBC Master microcomputer), or WORDWISE on a BBC micro. VIEW can be used, but the format and justify options should be turned off and you should do not create any new rulers nor enter any formatting commands (shift-f8). EDWORD files are not suitable, but spooled output from this editor is acceptable. For very short mod-files it would be possible to use `*BUILD`.

The file may have any name as MERGE takes the filename as a parameter. Typically you might have a mod-file for each class held permanently on the fileserver, so that you can make changes to an entire class (e.g. remove them when they leave) very easily.

There may only be one occurrence of any given username in the mod-file.

*Global* keywords are specified outside a user definition and take effect for all the following users up to the next mode keyword. There may be many global assignments following each mode keyword; the assignments to a particular keyword are not cumulative (e.g. `ACC="1"`; followed later by `ACC="2"` is not the same as putting `ACC="1 2"`). All global keywords are reset by each mode keyword.

*Local* keywords are specified after a username within curly brackets { } and only affect that username. Local keywords take precedence over global keywords and again are not cumulative (e.g. global `Flag="Pw"`; followed by local `Flag="Ns"`; does not give `Flag="PwNs"`).

Comments can be specified by inserting an `"&"` as the first character of a line; the rest of the line up to a CR (CHR\$13) or a LF (CHR\$10) is then ignored.

There is one important restriction on the size of the Mod-file, that is that it cannot contain more than 256 users. However, this should not present a problem as MERGE can be run as many times as necessary on different mod-files, without having to re-run CONVERT or GENERATE.

The general form of a mod-file is shown below :-

.mode.

```
Global assignments
Username { local assignments }
Username { local assignments }
Global assignments
Username
```

```
.mode.
Global assignments
Username { local assignments }
.END.
```

## Running \*MERGE

```
*MERGE
*Merge [<mod-file>]
Version 1.16, (C) SJ Research
Parses mod-file and produces T.Int1,
then Merges T.Int1 with T.Genfile
Also produces !makdir & !remdir
```

```
Parsing Mod-file....
Warning - !makdir already exists.
Warning - !remdir already exists.
(O)verwrite, (A)ppend or (Q)uit
O/A/Q ?O
Merging T.INT1 & T.GenFile.
***** Error : at line 1
TONY {}
-----
SYST - User not found
...
Errors during Merge - aborted
```

## Errors

During the *parse* stage, the line number of the line in error and a relevant portion of the mod-file will be printed. During the merge stage, the line number and relevant portion of T.GENFILE will be printed. Lines are numbered starting from 1.

The following keywords may be defined either globally or locally :-

ACC, BASE, BOOT, CREDIT, FLAG, LIB, PACC, PASS, URD.

The keyword DEFAULT may only be defined locally.

In addition there are the following modes.

.ADD. , .REMOVE. , .MODIFY. , .END.

Please note that changing mode sets all global assignments to their defaults.

### 5.5.3.1 !makdir and !remdir

N.B. To use either of these files, the user requires system privilege, key in the SYSTEM position, and ownership of all accounts.

MERGE always creates, in the currently selected directory, the files !makdir and !remdir. These files contain a sequence of commands which will make/remove directory structures and also to credit/debit accounts, for any users added or removed during the merge process. In .ADD. mode, commands to create

the relevant URD directory and to credit the relevant account are added to !makdir, in .REMOVE. mode commands to delete the users' URD and all its contents and to debit the account are added to !remdir. In .MODIFY. mode, no commands are added to either file. To use the files, type

\*EXEC !makdir or \*EXEC !remdir.

N.B. Once either file has been \*EXEC'ed, it should be deleted by using \*DELETE !makdir or \*DELETE !remdir.

MERGE can also append new information to the end of an existing file, so that a sequence of MERGEs can be done, followed by a single \*EXEC command.

The following !makdir file was created as a result of the mod-file in §5.5.6. The disc name is Work. Bold type indicates user input.

```
*EXEC !makdir
*DIR :Work
*CDIR CLASS87
*DIR CLASS87
*CDIR FRED
*DEBIT 145 65535
*CREDIT 145 100
*ACCOUNT FRED 145
*DIR :Work
*CDIR TONY
*DEBIT 10B 65535
*CREDIT 10B 50
*ACCOUNT TONY 10B
*DELETE !makdir
```

The \*CDIR commands are inserted unless the URD keyword was defined to be the root, i.e. "\$" or ";<discname>". If URD=""; the directory \$.<username> will be created. The \*DEBIT, \*CREDIT and \*ACCOUNT commands are included whenever the user has a Personal account number (i.e. PACC<>").

An example of the contents of a !remdir file:-

```
*BASIC
LOAD"ERAQ"
RUN
:Work.CLASS87.FRED
N
*DEBIT 145 65535
```

For !remdir, URD and PACC definitions are taken from T.GENFILE. The "ERAQ" sequence of commands (lines 3 to 5) is present unless the URD is defined to be the root of any disc, the \*DEBIT is inserted when the user has a PACC. If some of the commands in !remdir are undesirable, they may be removed/modified with a text editor.

## 5.5.4 \*GENERATE

Syntax: \*GENERATE  
System Privilege Required.

The key need only be in the SYST position during the actual installation of %PASSWORDS, i.e. only for the very last phase of the program. Ownership of the root on the relevant disc is also required (usually account 0).

The GENERATE program creates a new filesaver-format password file PASSWDS from T.GENFILE.

GENERATE then installs this as %PASSWORDS and the user should then delete T.GENFILE and turn the key back into the SECURE position.

The file PASSWDS is always created in the root of the disc onto which the new password file needs to be written. This is because the file PASSWDS is transferred to %PASSWORDS using a \*RENAME command. This has the advantage of being an 'atomic' operation, i.e. no operations from other users are allowed while it is happening (especially logging-on). It also automatically deletes the original file. [This is a special case whereby \*RENAME is able to rename a file 'on top of' another already existing file; this does not work in reverse, and you cannot \*RENAME the password-file back out again.]

**Note:** Whenever the message Output now in file \$<discname>.PASSWDS occurs, this file will exist (with access "PWR/") and contain sensitive information until it is successfully installed as %PASSWORDS. Therefore, if the file is not installed, you should delete it. The file will also remain if you abort the installation.

GENERATE will only ask you whether you wish to install the new file when no serious errors were detected, and that the contents of the file are only guaranteed to be valid when there are no errors/warnings given. When GENERATE prompts the user, pressing any key other than 'Y' or 'y' will abort. Note that aborting in this way will leave the file in the root.

Running \*GENERATE:-

#### **\*GENERATE**

Password file editing system - GENERATE  
Reads file T.GENFILE, writes file \$.PASSWDS  
Version 1.08

.END. found

Number of users =&0022

Finished :  
00 warning(s)  
00 serious error(s)

Output now in file :SMALL1.PASSWDS

Install new file as %PASSWORDS ? (Y/N) Y

Installed.

## **5.5.5 Keywords**

All keyword assignments have the form:

<Keyword>="<value>";

Note the double-quotes and semicolon, which must always be present.

**ACC** Defines the group account numbers given to a user (see also PACC). Each account number is a number (in hexadecimal i.e. 0 to 9 then A,B,C,D,E,F) in the range 0 to 7FF. Multiple account numbers are separated by a space. To specify a range of accounts the first account number is the lower range followed by the '>' character followed by the upper range of the account number.

Group account numbers in the range 100..7FF are actually allocated in blocks of 40(hex). That is, if any account from 100 to 13F is specified, all accounts 100..13F will be owned by the user.

In .MODIFY. mode, the characters '+' and '-' may be used in the definition. All account numbers following such a character add or remove accounts as appropriate. A single definition may include both characters, and they will be evaluated on a left-to-right basis. E.g. ACC="0>7F -10>6F + 30 32" would give be equivalent to ACC="0>0F 30 32 70>7F". A whole account block can be removed by specifying a single account; for instance, ACC="-100" would remove accounts 100..13F.

When used in .ADD. mode, '+' or '-' will cause the error Bad character. In .REMOVE. mode, whilst local assignments are parsed, no error will be caused.

For example

```
ACC="10 2C 30>FF 140";
```

will assign account numbers 10, 2C, 30 through FF as well as group accounts 140..17F.

ACC defaults to "" i.e. no accounts.

## BASE

This defines the base user root directory name to which the username is added. It will generally be defined in a Global assignment, although it can be defined in a local assignment. No "\$." prefix is required - see the URD keyword.

For example:

```
.ADD.
```

```
BASE="CLASS87";
```

```
TONY {}
```

will give TONY the user root directory "\$.CLASS87.TONY"

BASE will override a URD assignment if it is defined at a later stage. Default for BASE is "\$" (and overrides until a URD is defined).

## BOOT

This defines the boot option for a user. The number ranges from 0 to 3 and has the following meaning on the BBC computer :-

- 0 - No action
- 1 - \*LOAD !boot
- 2 - \*RUN !boot
- 3 - \*EXEC !boot

The default value of BOOT is 0.

## CREDIT

In .ADD. mode (and no other), this will set the amount of space, in K, that is CREDITed to the personal account number. The password file itself will not be affected by the value of this keyword, only the !mkdir file is affected.

The number is in decimal and in the range 0-65535. The default value of CREDIT is 100.

## DEFAULT

This takes either 0 or 1 as a parameter. "1" sets this user to be the default user, "0" means that the user should no longer be so. MERGE never produces any warnings about 'silly' uses of DEFAULT (e.g. using DEFAULT="0"; on a user who wasn't the default user anyway).

May only be defined as a local keyword, and defaults to 0.

## DISCNAME

This keyword is always present in T.GENFILE and never anywhere else. It gives the name of the disc from which the password file came, and is put there by \*CONVERT.

## FLAG

For this keyword the assignment of data is in the form of two letter combinations which are

as follows.

Pw password locked  
Sy system privilege  
Ns No short saves  
En Permanent \*ENABLE  
Nl Library only used for \*RUN commands  
Ro '\*RUN only' user

A user with this option enabled may use \*RUN and certain other \*commands. Also permitted are FScall #14 (Read disc info), FScall #16 (Read date/time), FScall #25 (Read FS version number), and FScall #65 (Read/Write misc info). All other commands will give the error message Who are you?

Al Auxiliary account locked

When this is set, the user is not allowed to change the auxiliary account of any file or directory under any circumstances.

X2 Reserved

See Editpass for more information (section 4.3, page 4-8)

In addition to this, in .MODIFY. mode either '+' or '-' may precede any flags to either add or remove options.

For example:

```
.MODIFY.  
TONY {FLAG="+SyEn-Pw"; }  
will take the existing flag options set for TONY and add the system privilege and permanent *ENABLE and remove the option for password locked. If neither '+' or '-' are used in .MODIFY. mode then the new assignment will override the old definition for the flag.
```

Default is Flag=""; i.e. Password/\*OPT4 not locked, Not System Privileged, Short Saves allowed, \*ENABLE required (for wild-card \*DELETE), Library used for all operations, Not 'Run Only', Auxiliary Account not locked.

**LIB** This sets the initial library directory for the user. Default is "", which means that the fileserver will select \$.LIBRARY on the lowest numbered disc (a hard disc drive, if you have one).

**PACC** This keyword defines a personal account number and is a hexadecimal number in the range of 1 to 7FF. If set to "" it means that no personal account number is allocated. When used in a local assignment that particular personal account number is given to the user. If the personal account number has already been allocated to another user (as a PACC) then a warning will be given.

In .ADD. mode, when PACC is used in a global assignment it assigns the next free account number greater than or equal to the one specified. That account is then marked as allocated so that the next user will get a different account number. The file T.Pmap contains a map of the currently allocated PACCs, and this file is read and updated when using this feature. To disable this feature, set PACC="" in another global assignment; in a local assignment you would PACC="344" to assign a specific account number.

For example:

```
.ADD.  
FRED { PACC="500"; }  
STU { }
```

TONY {}  
.END.

will allocate personal account 500 to FRED, personal account 100 to STU (assuming it has not already been allocated to some other user) and personal account 101 to TONY. If personal accounts 100, 101 and 103 have been allocated to some other users then STU would be assigned personal account 102 and TONY would be allocated personal account 104.

Default is "100", i.e. allocation will start from 100 (.ADD. mode only).

**PASS** This sets a user's password, the default password is "".

**URD** This sets the user's root directory, and overrides any BASE definition. The fileserver selects the URD relative to the root of the disc on which %PASSWORDS is. Therefore you do not need to prefix it by "\$." unless the directory required is on a different disc (when you should use :<discname>). By default, BASE is set to "\$." and the URD is undefined (that is, it is not referenced). If URD is set to "", the URD becomes the default, i.e. \$.<username>. To set the URD to null, use URD="\$".

## Mode Keywords

**.ADD.** In this mode the user entries are taken as new users. If a user of this name already exists an error is generated. A set of commands are placed in the file "!mkdir" to create the appropriate URD and credit the appropriate personal account (the system manager will \*EXEC !mkdir at a later stage). If MERGE is used repeatedly, new commands will be appended to the existing !mkdir file, and a warning will be given. Therefore, once !mkdir has been \*EXECed it should be deleted.

If a user has PACC set to "" then the account number of the URD created will be the account number of the parent directory (i.e. no \*ACCOUNT command will be placed in the !mkdir file). In this case it is possible that the user will not have owner access to his URD.

**.REMOVE.** The specified users are removed from the password file. Obviously no global assignments or local assignments are needed, however it is not an error for these to exist. This makes it possible to remove blocks of users and later restore them just by changing the mode keyword to .REMOVE. The curly brackets {} must be present after each user name, although there needn't be any text within them. However, the text inside curly brackets is parsed, so don't put garbage in there!

A set of instructions is placed in the file "!remdir" to delete the appropriate URD and its contents and also to debit all the space allocated to that account.

**.MODIFY.** The data in the user entries is used to modify the data already held in an existing entry. It is an error for the user not to already exist. To add or remove accounts or flags from a user entry the characters '+' or '-' may be used.

**.END.** This signifies the end of data in both the mod-file and the gen-file. The use of .END. is optional, but \*CONVERT always puts a .END. at the the end of the gen-file.

## 5.5.6 Mod-file Examples

Consider the following mod-file :-

.ADD.

```
ACC="1 23"; BASE="CLASS87";
FRED {PACC="145";}
TONY { URD=""; CREDIT="50";}
.END.
```

The mode keyword `.ADD.` specifies that the users are to be added to the password file.

The next line is a global assignment: the keyword `ACC` is assigned the values 1 and 23 and the keyword `BASE` is assigned the name `CLASS87`; as these appear outside a username definition they are global assignments.

The username `FRED` has a local assignment defining his Personal Account Number as 145, so he will have access to Accounts 1 & 23 (from the global assignment) and Account 145, and his default directory after logon will be `$.class87`. `TONY` has the keyword `URD` defined locally which overrides the global `BASE` assignment.

User `TONY` will have a `URD` of `$.TONY`, and will have a personal account number allocated (the lowest free one above 100). He will also have access to accounts 2 and 23, but will only have 50k of disc space allocated to his personal account.

The following mod-file has exactly the same effect as the previous example.

```
.ADD.
TONY { URD=""; ACC="1 23"; }
FRED { ACC="1 23"; URD="CLASS87.FRED"; PACC="145"; }
.END.
```

A typical application of the batch mode editor would be to add a new year's entry to the system.

Suppose we have a mod-file called `CLASS4A` thus:-

```
.ADD.

PACC="200";
BASE="Class4a";
CREDIT="50";
FLAG="PwNsAl";

ArdleighW {}
BassetF {}
MunroeM {}
BunterW {PACC="";}
WilliamJ {}
BottVE {}
KermitF {}
BigglesDSO {}

.END.
```

By typing `*CONVERT`, `*MERGE CLASS4A`, `*GENERATE` you will now have an updated password file installed. To create the necessary directories, type `*EXEC !mkdir`. That's all there is to it.

You would normally keep the file `CLASS4A` around; in order to delete the users, change the `.ADD.` keyword to `.REMOVE.` and repeat the process, only this time finish off with `*EXEC !rmdir`.

## 5.5.7 Warnings and Errors

Warnings and errors are accompanied by a portion of the relevant file and two inverse exclamation marks

(\*CONVERT/\*MERGE) or a left-pointing arrow (\*GENERATE) to indicate the approximate location of the error. N.B. in mode 7, these symbols appear as white squares (character 255).

### **\*CONVERT**

URD/LIB pointer corrupt for user - <username>  
EOF (no terminating user entry, or password file corrupt)  
Bad disc name  
%PASSWORDS not found  
Directory called T not found  
\*\_\*-\*\_\* System Error : <OS error message>

### **\*MERGE - Parse stage**

>256 users in mod-file  
.END. is a global keyword  
Bad character  
Can't find mod-file  
Can't find T.pmap  
Default is only valid as a local keyword  
End of file inside quoted string (or missing ")  
Expecting a "  
Expecting a }  
Expecting a number  
Flag not known  
Keyword/userid too long  
Mode keywords must be specified globally  
Mode not specified, using .modify. by default  
Need an = to assign value  
No more personal account numbers!  
Number too large  
Parameter too long  
Personal account number already allocated  
T.pmap has not been generated by \*CONVERT  
Text found after .END.  
Unknown keyword  
User already exists (two users of same name in the mod-file)

### **\*MERGE - Merge stage**

Bad keyword in Genfile  
Flag not recognised  
No users found in Mod-file!  
Second number in range smaller than first  
T.Genfile not found  
<Username> - User already exists in password file (in .add. mode)  
<Username> - User not found (in .modify. or .remove. modes)  
Warning - !mkdir already exists  
Warning - !remdir already exists

## **\*GENERATE**

### **Fatal errors:**

File not found - T.GENFILE

### **Errors:**

Discname too long

Mismatched { } brackets

Larger number of users in pass 2 - output file useless

Two users with same name, or not in alphabetical order

Userid is missing/zero length

\*\_\*\_\*\_\* System error : <OS error text>

### **Warnings:**

Bad number

Bad operator - expecting "=" or "{"

Bad range (second number after range indicator wasn't a number)

URD text exceeds 80 characters

BASE text exceeds 80 characters

Boot option >3

Constructed URD exceeds 80 chars

DEFAULT cannot be used as a global keyword

DISCNAME must be global keyword

Keyword/userid too long

Missing " in assignment to keyword

More than one default user - this one ignored

Odd number of characters in FLAG text

Password exceeds 10 characters

Significant text after .END. - ignored

Smaller number of users in pass 2

Start of range bigger than end

Unrecognised flag name

Unrecognised keyword

## **5.5.8 File Specification**

All characters ASCII 0 through ASCII 31 are considered as a SPACE. Top bits are stripped. There is no case sensitivity, as every alpha-numeric is taken as upper case.

<file> ::= <gen-file> | <mod-file>

<gen-file> ::= DISCNAME="<discname>"; <pw data> .END.

<mod-file> ::= .<mode>. <pw data> [mod-file] [.End.]

<pw data> ::= [<userdata> | <global assignment> | <comment>] [<pw data>]

<mode> ::= Add | Modify | Remove

<comment> ::= & <text> <line terminator>

<line terminator> ::= <CHR\$13> | <CHR\$10>

<global assignment> ::= <global keyword> = "<keyword value>";

<userdata> ::= <UserID> { [<local assignment>] } [<userdata>]

<local assignment> ::= <local keyword> = "<keyword value>"; [<local assignment>]

<UserID> ::= [<alphanum>]  
 <keyword> ::= ACC | BASE | BOOT | CREDIT | FLAG | LIB | PACC | PASS | URD  
 <global keyword> ::= <keyword> | DISCNAME  
 <local keyword> ::= <keyword> | DEFAULT  
 <keyword value> ::= <acc> | <lib> | <pass> | <boot> | <urd> | <flag> | <pacc> | <default> | <base>  
 | ""  
 <acc> ::= <hex> | <hex> > <hex> | -<acc> | +<acc>  
 <lib> ::= <path>  
 <pass> ::= <alphanum>  
 <boot> ::= 0 | 1 | 2 | 3  
 <urd> ::= <path>  
 <pacc> ::= <hex>  
 <default> ::= 0 | 1  
 <base> ::= <path>  
 <flag> ::= [<flagsymbol> | +<flagsymbol> | -<flagsymbol>]  
 <flagsymbol> ::= Pw | Sy | Ns | En | NI | Ro | Al | X2  
 <path> ::= <name>[.<path>] | :<discname>[.<path>]  
 <discname> ::= <alphanum>  
 <name> ::= <alphanum>  
 <hex> ::= <hexit> | <hexit><hexit> | <hexit><hexit><hexit>  
 <hexit> ::= 0|1|2|3|4|5|6|7|8|9|A|B|C|D|E|F

## 5.5.9 Known Problems

### General:-

Whilst all programs set protection against remote network operations (\*VIEW etc) during operation, the current versions of CONVERT/MERGE do not clear RAM before exiting (GENERATE does), although they do leave the computer protected after exiting. Therefore, you should always logoff and then switch the computer off (the order is important) after you have finished using these programs.

### \*CONVERT, version 1.12:-

A corrupt %PASSWORDS file (with no terminating user entry) will give an EOF error. The resulting T.GENFILE will contain useful information, but T.PMAP will not have been saved. You should \*TYPE T.GENFILE to find out whether most of the users have been included; there may be some corrupt users at the end of the file - these should not matter. Then, using \*GENERATE (which does not require T.PMAP) you can create a repaired password file, which can then be re-CONVERTed correctly.

Some fatal errors (e.g. Account Bankrupt, Disc full, and Network errors) cause CONVERT to abort without closing files.

If the DEFAULT USER pointer is corrupt, CONVERT will not produce a warning; no user will have DEFAULT="1" in the gen-file.

### \*MERGE, version 1.16:-

\*MERGE does not give a warning when more than one user has been assigned DEFAULT="1". GENERATE will however give a warning, and will ignore subsequent assignments. In the case where DEFAULT="0" is specified when the particular user was not already the default user, again no warning will be given.

High-numbered group accounts do not act as blocks e.g. if you have ACC="100>17F" in T.GENFILE, the do ACC="-140", the result will be ACC="100>13F 141>200". GENERATE however does treat them in blocks, so will give access to A/cs 100>200 as before. Actually this is not normally a problem, since CONVERT does not produce ranges for high numbered accounts (ranges are only produced by MERGE), it merely specifies the base account number. From a password file it would have given ACC="100 140" whereupon removing account 140 using MERGE would have had the desired effect.

### \*GENERATE, version 1.08:-

Does not give a warning if there are no system privileged users with access to account 0.

# Chapter 6: Setting up printers

---

## 6.1 Overview

SJ Research File Servers all contain a built-in *printer server* with facility to connect one or more printers. One or two physical printers may be connected to the Modular Disc File Server: one to the parallel output and one to the serial output. The connection of printers is described fully in Appendix C.2.

Usually a *banner* will be printed before each user's output: this is some text set up by the system manager, which may also contain information (such as user identifier, time, date etc.) inserted by the Printer Server.

An example of a banner is:

```
SJ Research File Server *** Station 5 (FRED) 08feb85 15:21:04 ***
```

The banner file also specifies some text to be added at the end of each user's printout. An example may be a row of asterisks followed by a page throw.

Each *physical printer* may have up to four different banners available, and these are distinguished as different *logical printers*. Thus there may be up to eight logical printers on each SJ Research printer server.

**\*PSLIST** will list all the printer servers and logical printers available on the network. The system manager has to give names to the logical printers connected to the File Server, using the program **Editprint** (see 6.3.1 below).

Directing output from a BBC computer to the network printer is quite complicated. In a simple system, where there is only one Printer Server and only one printer, **\*PS** will select it. In most cases, **\*PS <name>** will be required to select a particular printer by name.

For detailed control of printing, there are three settings to be adjusted : whether your computer uses a directly connected printer or a network printer, which printer server *station* is used, and which *logical printer* is selected within that station. The first two of these are remembered by your BBC micro, while the logical printer setting is remembered (separately) in each printer server. **\*PS**, **\*PRINT**, **\*PRINTER** and **\*FX5,4** can all be used to adjust different combinations of these settings.

When first switched on, a BBC micro will assume that a directly connected parallel printer is to be used, and if the network printer is then selected it will use the printer server at station 235. The default logical printer can be adjusted by use of *Editprint* (see 6.3.1 below). Remember that if the equipment is left switched on, any changes will be remembered indefinitely, so it is not wise to rely on these default settings.

**\*PS** alone will choose (at random) any printer server station that has a working printer connected, but will not change the logical printer setting. **\*PS <name>** will select a printer server station with a printer of that name and select that logical printer in all printer servers that have one. **\*PS <number>** will select that printer server number regardless of whether it has any printers connected, and will have no effect on logical printer settings. Hence **\*PS** is useful if there is only one possible printer for it to select, **\*PS <name>** is the most commonly used variant, and **\*PS <number>** can be useful if there are several printers with the same name. All variants of **\*PS** also have the effect of **\*FX 5 4** and possibly **\*FX 6** - see below.

**\*PRINT** changes the selected printer server station number to that of your currently selected File Server. The logical printer setting is not changed: if there is more than one logical printer available, **\*PRINTER** can be used to select between them. **\*PRINT** also has the effect of **\*FX 5 4** and **\*FX 6**.

**\*FX5,4** instructs the BBC computer to use the network rather than a directly connected printer. Neither the printer server station number, nor the logical printer name are affected by **\*FX 5**.

**\*PRINTER <name>** adjusts the logical printer setting *in the printer server attached to the currently*

*selected files server only.* It has no effect on any settings in the BBC computer. It is therefore useful in conjunction with **\*PRINTOUT**, which can only use the current files server, or **\*PRINT** which forces the BBC computer to use the current file server as printer server. **\*PRINTER** alone will display the current setting without changing it.

In summary, **\*PS** is the most useful command for general printing from a BBC computer, as it sets all three pieces of information. The other commands are available for fine control in very complex situations.

The **\*PS** command actually loads a program called PS from the library directory. There are two versions supplied, called **PSFX6** and **PSnoFX6**. If your printer does an automatic line-feed after every carriage return, then use **PSnoFX6**, otherwise use **PSFX6**. The latter does the call **\*FX6,0** which allows the BBC computer to send line-feed characters to the printer. When first supplied, the library has **PSfx6** under the name **PS**, and **PSnoFX6** under its own name. If your printers are set for auto line-feed, type:

```
*DIR $.library
*RENAME PS PSfx6
*RENAME PSnoFX6 PS
```

to make the correct version available to your users. The utility **\*CHECK** can be used to tell which version is currently named **PS**.

If you have two printers, one of each type, use **PSnoFX6** as **PS**; and instruct users to type **\*FX6,0** after **\*PS** if they have selected the printer with automatic line-feeds. Alternatively, see if you can disable the automatic line-feed on the printer, so that you can then use **PSFX6** throughout.

### Special note for BBC Master Series computers

BBC Master Series computers (Master 128, Master ET, Master Compact etc.) have the ANFS ROM fitted. This means that there is a version of **\*PS** built in to the computer, and that the default settings which were fixed on the BBC computer can now be configured by the user.

The version of **PS** built in to the ANFS is broadly similar to the version supplied on the files server disc, but the messages it produces are different and *it does not set \*FX5,4 or \*FX 6*. The easiest solution is to always type **\*/PS** (rather than **\*PS**): this will use the standard version from disc, regardless of whether it is typed on a standard BBC computer or a Master Series computer.

Alternatively, the built-in **\*PS** can be used, and the values of **\*FX 5** and **\*FX 6** set up permanently in the CMOS RAM.

To set **\*FX 5 4** as default:

```
*CONFIGURE PRINT 4
```

To set **\*FX 6 0** as default:

```
*CONFIGURE IGNORE
```

Alternatively, to set **\*FX 6 10** (equivalent to **PSnoFX6**):

```
*CONFIGURE IGNORE 10
```

To set the default printer server station number:

```
*CONFIGURE PS <ps number>
```

Remember that there is no protection against any user re-configuring these settings, so in situations where many people use the same computer it is easier to use **\*/PS** rather than relying on the configured settings. Note also that newly supplied Master Series computers will probably have these settings configured inappropriately.

## 6.2 Everyday printer management

While it is hoped that users will send listings of programs and text from wordprocessors it is likely that, by accident, a user will select and start printing to the network printer when he does not intend to. If the printer data is being spooled to disc then the user may not realise that he has a <Ctrl B> active. This will lead to a print job being held in the %PRINTQ directory, taking space on the disc.

If the user is aware that he has accidentally sent spurious text then a **\*FLUSH** can be used to delete all print jobs whose username and station number match those of the user, whether those jobs are currently printing or not. For the printer managers (users who have access to the main account number of the %PRINTQ directory) there are additional commands to control the printers:

### **\*PSTOP [<printer name>]**

This command suspends printing on the specified printer (or both printers if no name is supplied). Any jobs which are currently printing are not deleted, but the corresponding file in the %PRINTQ is closed, and may hence be manually deleted if necessary. If printing is restarted without changing anything, the same job will be re-printed from the beginning, which may be useful after a paper jam has been cleared.

### **\*PGO [<printer name>]**

Restarts printing on the specified printer after it has been suspended by **\*PSTOP**. Both printers are restarted if a name is not supplied. **\*PGO** has no effect on printers that have not been suspended. Note that printing will also be restarted on both printers by the *Save changes & exit* option in **Editprint**, and when the fileserver is restarted. Hence if the printers are accidentally suspended, they may unexpectedly start printing the next time that the fileserver is restarted or the discs are changed.

%PRINTQ is a directory on the lowest numbered disc on the File Server. It can be accessed by name like any other directory but does not appear on the catalogue of the root directory. %PRINTQ need not have a pathname given for it, even when it is accessed from another disc.

Each item waiting to be printed is transferred to %PRINTQ and given a unique name, starting from AA00 and going up to ZZ99. Then, when a printer becomes free, the directory is checked in alphabetical order for the first job waiting to use that printer. When each job has been completed, its entry is erased from the print queue and the next job is printed. Thus, by default, print jobs to a particular physical printer are carried out in the order in which they are submitted.

The account number for each print job is reset to that of the %PRINTQ directory, so that users' own account credit is not used up. However the auxiliary account number of a print queue entry is set to the user's personal account number. This usually means that a user has owner access to his own print jobs, so he has read access to them and can also delete them from the print queue.

Information about entries in the print queue can be obtained with the usual directory commands **\*CAT**, **\*INFO** and **\*EX**. Print queue entries have a special access code /spl, which indicates their status as print jobs. Entries submitted by **\*PRINTOUT** (see Section 6.5) will have access code /prt. Like directories (which have D/), these access letters cannot be changed by the **\*ACCESS** command, but it is possible to add **P** or **L**. Unlike ordinary files, the **L** access letter does not prevent the file from being deleted by the user, but it does prevent the system from printing the file.

The formats of **\*EX** and **\*INFO** are also changed for print jobs, so that the user and station number originating the print request is given in place of the load and execute addresses of a file, and the logical printer selected in place of the creation date. For example, **\*EX %PRINTQ** might give

```
%PRINTQ (126)  Public
Main-V        Option 03 (Exec)
Dir. Karen    Lib. Library
AA33  DIANA   at Stn.005 00008A /spl HOLD    13feb86 14.32 01 (FF)
AA34  TONY    at Stn.064 00DC43 /spl PARALL today 11.59 01 (F0)
```

There are three special logical printer names: **HOLD**, **AUTO** and **PRINT**. Jobs sent to the **HOLD** printer are not printed: they remain in the print queue until they are either deleted or rerouted to a real printer.

**AUTO** automatically selects whichever printer is free, in an order of preference set up by **Editpass**. Both **AUTO** and **HOLD** operate like any other logical printer: they can be specified in **\*PS** or **\*PRINTER** and jobs can be sent to them by **\*REROUTE**. **PRINT**, however, is a special name for use in **\*PS** which leaves the logical printer setting unchanged. **PRINT** will never appear on jobs in the print queue: they will be marked with the logical printer setting which was actually in force.

Printer managers (who own the *account number* of the **%PRINTQ** directory) will have owner access to all entries, and will be able to read, rename and delete any of them. Because the queue is sorted into alphabetic order, changing the names of files in the print queue will change their relative priorities. The printer manager can thus "queue-jump" important jobs to the head of the print queue by using **\*RENAME** and some suitable name like **!URGENT**. (! has highest alphabetic priority, followed by the numbers 1 to 9 and then the main alphabet.) Naturally enough ownership of the print queue account should only be given to trustworthy people like the system manager.

When the File Server is re-started, the next name for a print queue entry is reset to **AA00**. If it is important that entries from the stored print queue are printed first, they should be renamed to restore their original priority. This is most easily done with the command:

**\*RENAME %PRINTQ.\* %PRINTQ.1\***

before any new entries are added to the print queue.

Privileged users can also create ordinary files and sub-directories in the print queue directory if they so wish. While this will not affect the print queue operation, it is not recommended.

The printer manager will be able to selectively remove jobs by **\*DELETE**. If the job is currently being printed then it is necessary to use **\*PSTOP** otherwise the error **Already open by 2.000 \*-SPOOL-\*** will be returned.

In order to remove spool files that have been opened by users with **<Ctrl-B>** who have then gone away or switched off their machines it is necessary to log them off before attempting to delete their output. The easiest way of doing this is to find the station number of the offending spool file by use of **\*INFO** or **\*EX** and then log them off using **\*LOGOFF <station number>**; you will then be able to delete the entry. It is advisable to reroute such jobs to the **HOLD** printer before logging off the offending user, as the job might otherwise begin to print.

Note that **\*LOGOFF** when used with a station number is a system privileged command, so you will have to be logged on as a system user to use it. Alternatively, **\*LOGOFF** can be typed at the station in question, which does not require system privilege. (**\*LOGOFF** is more powerful than **\*BYE**: **\*BYE** does not fully log off a user if the file server thinks that they are still printing, until a **ctrl-C** is received).

## 6.3 Setting up the Printers

The File Server has to keep a number of pieces of information about the printers connected to it. The electrical parameters of the serial printer (baud rate etc.) are kept within the fileserver itself, but all the other information is kept on disc. Most of the settings are kept in a special part of the disc which can only be accessed by the **Editprint** program. If you are using more than one disc, this information will be kept on the first disc in the system (i.e. the first in the list produced by \*FREE). This will be a hard disc if you have one, but will otherwise be the floppy disc in drive A. If you use different floppy discs in this drive, remember that you will need to use **Editprint** on each of them. The **Editprint** information is copied by the *copy discs* option in utility mode.

Further information is kept in *banner files*, but these are ordinary files which require no special precautions.

### 6.3.1 Editprint

Editprint is a BASIC program, which can only be used by system privileged users.  
To adjust the printer settings, type:

```
CHAIN "EDITPRINT"
```

The program will respond with:

```
Edit logical printer details
Change system messages
Set up initial choices
Save changes and exit
```

Choose an option by moving the menu bar, with the cursor keys, over the option and pressing return.

#### Option 1 - Edit logical printer details

This will result in a list of *logical* printers being displayed on the screen, for example:

1.Microl	Parallel
2.Serial	Serial
3.Nobann	Parallel
4.	Serial
5.conden	Parallel
6.Epson	Serial
7.	Parallel
8.	Serial

The right hand column indicates which *physical* printer will be used: while the File Server only has connections for two printers, it is possible to have several printer names associated with each one.

By moving the menu bar and pressing return, an individual logical printer can be selected, and its details will appear:

```
Name: MICROL
Printing enabled: Yes
Bannerfile: Banners.Parallel
Spool to Disc: Yes
Anonymous Users allowed: Yes
Account Ownership required: No
```

Again the menu bar highlights one item. Yes/No items can be changed by pressing **space**, while other items can be changed by typing a new value, followed by **return**. **Return** on its own writes any changes back to the File Server and returns to the main menu. **Escape** discards any changes which may have been made by mistake.

**Name** is the name which users will quote to specify that particular logical printer. Printer names may be up

to six characters long. The names PRINT, HOLD and AUTO are reserved and must not be given to a printer. If the printer name is blank (i.e. consists of spaces), that printer is disabled completely.

**Printing enabled** controls whether output sent to this particular printer will be printed. It does not prevent users from generating output, which will be spooled to disc. Hence it is possible to have two logical printers named 'PAPER' and 'LABELS', only one of which is enabled at any time. Users can generate both types of output, and any documents sent to the disabled printer will be held until someone changes the stationary in the printer and uses EDITPRINT to enable the corresponding printer name. Another use for printer names with printing disabled is to allow users to generate output for a *remote despooler* program: this ensures that the File Server itself does not try to print jobs intended for a distant printer.

**Banner file** gives the name of a text file which controls the *banner* that is printed at the top of all printer output. The various possibilities for the contents of the banner file are described in section 6.3.2. The file name is looked up starting from \$ on the first disc drive, so **banners.fancy** would be equivalent to **\$.banners.fancy**. If the file cannot be found, or if the name is blank, no banner is printed at all: this is useful for non-standard devices such as graph plotters. Note that the system must have read access to the banner file: the access string on the file would usually be set to **WR/**.

**Spool to Disc** controls whether printing starts as soon as some data arrives, or whether it is spooled onto disc and only printed when the whole document has arrived. In either case, data will be spooled to disc if the printer is already busy with another user's output. For fast printers, it is preferable to spool to disc, preventing one user from claiming the printer for an extended period. For slow printers, or graphics dumps, it saves time to start printing immediately.

**Anonymous users allowed** control whether users who have selected this logical printer, but are not logged on to the File Server, may print. If this user presses <CTRL B> then he will be logged on as **ANONPRINT** or the default user if ANONPRINT does not exist. Having finished printing the user will be automatically logged off.

**Account ownership required** controls whether a user requires a specific account number to select this logical printer, beware that if this printer is listed under initial choices then the account ownership check will be bypassed.

## Option 2 - Change System Messages

This enables you to set the level of system messages. If you select option two from the editprint menu the following will appear on the screen:

```
0 = serial
1 = parallel

System message   Parallel
Message Level is 0
```

N.B. System error messages are ALWAYS sent to the printer.

By typing zero or one, the printer port used for system messages can be selected. It is not possible to disable system messages altogether, as the system has to have some way of displaying warnings of impending failures.

The system message printer should be one which is usually connected, or if there is usually no printer connected, the type of printer which can most readily be found in emergency should be selected. Note that the screen of a BBC micro can be used as a serial terminal if a suitable lead is available.

If the menu bar is moved down, a list of the possible message levels appears. The selected level can be changed by typing a number followed by **return**.

```
0 = off
5 = logon/logoff
```

```

7 = errors
10 = maximum users & *commands
11 = load/save
15 = *cat and opens
128 = aborted loads
130 = Fn codes
150 = net errors
170 = map building
200 = disc read/write
250 = all sucessful net transactions
255 = all activity to JPROC

```

```

System message parallel
message level is 0

```

These represent cummulative levels of system messages (7 includes the message of 5 and 0). Although the system message level may be set to 0, system messages after catastrophic errors will still appear on the printer. The printer specified can also be used for user's output, since the system messages will be separated by a page throw and header. If the message level is not 0 then any corresponding logical printers should be set to *not enabled*. This will prevent users from printing to them.

The usual message level is zero.

### Option 3 - Set Up Initial Choices

This option allows you to specify the default printer for users who do not select a particular printer, and to indicate the first and second choices of printer when **AUTO** is selected as a printer.

The screen will display:

Priority of Printer	No	Printer
	0	STOP
1st Microl	1	MICROL
2nd Epson	2	SERIAL
	3	Nobann
	4	
	5	conden
	6	Epson
	7	
	8	
Default AUTO	9	HOLD

```

New default choice.
Press 0-9 to select printers

```

Use the up and down cursor keys to select the piece of information you wish to change and enter the required printer number. The display on the right hand side of the screen lists the available printers. If the second choice is set to zero, then **AUTO** will be equivalent to the first choice printer. If the first choice is set to zero, the **AUTO** printer will be disabled completely.

Remember that any user can select and print to the **AUTO** printer, bypassing any restrictions that may be placed on the first or second choice printers. Hence it is not usually sensible to specify as first or second choice a printer which has account ownership required.

It is conventional to set up the **AUTO** printer such that the first choice is the fastest printer for long listings, with the second choice being the other printer with a similar banner. If the second printer is unsuitable for listings, the auto printer would usually specify just one choice, or be turned off altogether.

The **default** printer is the one which will be used if a user sends data for printing without selecting a

particular printer. The File Server still checks whether the user is permitted to use that printer, so restricted access on the default printer will prevent some users from printing without explicitly selecting a printer to which they do have access.

Popular settings for the default printer are 0 (AUTO) or 9 (HOLD).

#### Option 4 - Save Changes And Exit

This option puts into effect any changes which have been made through the other **Editprint** options. Note that the changes have already been written to disc, so leaving **Editprint** without using this option will not discard the changes: they will come into effect next time the fileserver is re-started.

### 6.3.2 The Banner File

The printer server usually prints a heading at the start and end of each piece of printed output, known as the *banner*. The name of the file to be used is set up with the Editprint program (see section 6.3.1). The format of the banner is controlled by a file associated with each printer name, and it may contain both fixed text and some information about that job, such as the name of the user that generated it and the date. It is possible to specify the same banner file for all the printers, but it is often useful to have more than one banner. In particular, it is possible to have two or more printer names which specify the same printer but with different banner files: a printer name **NLQ** might specify a banner file containing the necessary control codes to set that printer into near letter quality mode, while there would be another printer name **DRAFT** which used the same printer but left it in draft mode.

If no banner file has been specified, or the file specified is not found, or the system has not got read access to it (this means that the file must have letter **R** owner access), then the users' text will be printed without a banner or endtext: no error message will be produced.

The disc supplied with the fileserver has a directory **\$.BANNERS**, containing (initially) two sample banner files, called **SIMPLE** and **FANCY**. The banner for each logical printer is initially set to no file, and the system manager will have to select one of these two files, or create one of his own, in order to get printer banners at all.

It is possible to allow selected users to change the banner for their own purposes, by giving them access to the account of the banner file (preferably having set this account to a unique value, so that the users in question cannot change any other files). It is recommended that the user uses the **\*RENAME** command, to change the name of the banner file to something else, and again to rename the desired file as the banner file: this avoids the chance of accidentally deleting the main banner file. The users who are allowed to change the banner file will have to be responsible themselves for changing it back after they have finished.

The banner file is a simple text file, of the sort created by **\*BUILD** or **Wordwise**. It contains a mixture of straightforward text, which is just printed out, and special symbols which are replaced by the information they represent. The file split into two parts: the *banner* which is printed at the top of a user's output, and the *endtext* which is printed at the end, separated by the special symbol **<BANNER>**.

end-text<BANNER>banner text

The special symbol **<BANNER>** must be enclosed in angle brackets as shown. Note that the end-text (the characters to be printed after the user's output) comes first, and the banner text itself after the word **<BANNER>**. The texts will be printed literally until a special symbol is encountered.

The banner or end-text may contain special symbols from the list below. Note that all the symbols are enclosed in angle brackets **<>**. A carriage return character in the text will cause a carriage return on the printer; note that if your printer does not do an automatic line-feed after each carriage return, then a line feed character (or **|J**) should be put after each carriage return character (unless you intend to over-print lines).

There are three symbols that do not cause anything to be printed directly, but they select which of three possible times are printed when the identifiers <HOURS>, <MINUTES> etc. are used.

- <NOW> selects the current time of day, at the moment when printing is actually taking place. The time to be printed is frozen at the instant when the <NOW> symbol is processed: this avoids inconsistent results if the clock ticks between the printing of the hours and minutes. If another <NOW> is encountered (in the end-text, for example), this will freeze a different value, as time will have elapsed during the printing of the intervening text.
- <START> selects the time of day at which the printing job was initiated, from the user's computer. Note that, especially when print spooling is used, this (and <END> below) may be substantially earlier than the time given by <NOW>.
- <END> selects the time of day at which the user finished sending characters to the printer.

The following identifiers cause part or all of the time and date as selected above, to be entered into the banner or end-text string. The default time selection is <START>.

- <HOURS> gives a two digit hour in the 24 hour clock, with leading zero printed. For example 6 pm will be printed as 18.
- <H> is a synonym for <HOURS>.
- <MINUTES> gives the minutes past the hour as two digits between 00 and 59, with leading zero printed.
- <M> is a synonym for <MINUTES>.
- <SECONDS> gives the seconds past the minute as two digits between 00 and 59, with leading zero printed.
- <S> is a synonym for <SECONDS>.
- <12HOURS> gives the hour in the 12 hour clock, with leading zero replaced by a space.
- <AM> gives am or pm as appropriate. Note that noon is deemed to be pm.
- <DATE> The day of the month, as two digits, with leading zero replaced by a space.
- <ST> gives the correct suffix to the day of the month. For example, on the first of the month, the string inserted by <ST> will be st, on the second, nd, on the fourth th and so on.
- <MONTHNAME> gives the full name of the month, e.g. **January**. No leading spaces are printed.
- <MTH> gives a three letter abbreviation of the name of the month, beginning with a lower case letter. January will be printed as **jan**.
- <MONTH> gives the number of the month as two digits with leading zero printed. January will be printed as **01**, and December as **12**.
- <YEAR> gives the last 2 digits of the year. 1987 will be printed as **87**.

The remaining symbols allow the banner to print the users name and station number and deal with the layout of the banner.

- <USERNAME> gives the user identifier logged on at the station that originated the print job. No leading spaces are printed.
- <STATION> gives the number of the station that originated the print job. The station number is printed with leading zeroes and with the network number (if the station was on a different network), but no leading spaces are printed. For example, station 2 on the local network will be printed as 002, but station 43 on network 7 is printed as 007.043.
- <BANNER> The delimiter between the *end-text* (which should appear in the file first) and the banner proper. See description above for full details.
- <B> is a synonym for <BANNER>.
- <MARK> gives a reference point for <TAB> (see below).
- <TAB nnn> pads out to a position *nnn* spaces from the last <MARK> identifier. There must be one space (only) between the word TAB and the number. If no <MARK> has been given, this command pads out to a position *nnn* spaces from the beginning of the text. Note that a carriage return does not reset the value of <MARK>, and that only the least significant byte of *nnn* is read. <TAB 0> is illegal: the instruction will be ignored and the word <TAB 0> will be printed. If the number after TAB is less than the current character position, then the tab will move to the position 256+*nnn*.

Note that all the special symbols are enclosed in angle brackets <>. Unrecognised special symbols will be printed literally.

Control characters may be sent to the printer either by direct inclusion in the banner file (if your editor allows this), or by use of the 'I' character:

- | introduces a 'control' character, e.g. |A inserts <ctrl-A>.
- ? inserts ASCII character &7F (delete).
- ! inserts the next character with &80 added (i.e. top bit set).
- < or > inserts characters < or >.
- || prints the 'I' character itself.

### 6.3.2.1 Table of standard characters

The following table shows, for each possible character code, a sequence of characters that can be inserted in a banner file to produce that code. In all cases except characters 60, 62 and 124, the same effect can be achieved by inserting the character values directly: the advantage of using these sequences is that the resulting banner file can be inspected with **\*TYPE** or a standard text editor.

0	@	27	[	54	6	81	Q	108	l
1	A	28	\	55	7	82	R	109	m
2	B	29	]	56	8	83	S	110	n
3	C	30	^	57	9	84	T	111	o
4	D	31	_	58	:	85	U	112	p
5	E	32	<SPACE>	59	;	86	V	113	q
6	F	33	!	60	<	87	W	114	r
7	G	34	"	61	=	88	X	115	s
8	H	35	#	62	>	89	Y	116	t
9	I	36	\$	63	?	90	Z	117	u
10	J	37	%	64	@	91	[	118	v
11	K	38	&	65	A	92	\	119	w
12	L	39	'	66	B	93	]	120	x
13	M	40	(	67	C	94	^	121	y
14	N	41	)	68	D	95	_	122	z
15	O	42	*	69	E	96	£	123	{
16	P	43	+	70	F	97	a	124	
17	Q	44	,	71	G	98	b	125	}
18	R	45	-	72	H	99	c	126	~
19	S	46	.	73	I	100	d	127	?
20	T	47	/	74	J	101	e	128	@
21	U	48	0	75	K	102	f	129	A
22	V	49	1	76	L	103	g	.	.
23	W	50	2	77	M	104	h	.	.
24	X	51	3	78	N	105	i	.	.
25	Y	52	4	79	O	106	j	254	~
26	Z	53	5	80	P	107	k	255	?

### 6.3.2.2 Creating the Banner File

To create suitable files, there are 3 possible methods:

1. Use the **\*BUILD** command (documented in Section 6.6). This is the simplest method, but does not allow a single line with more than 255 characters. If this is going to be a problem, then use method 2 or 3.
2. Write a short BASIC program that calls **\*SPOOL <file name>**, then outputs the required text using **PRINT**, then closes the file using **\*SPOOL** on its own. For example:

```

10 *SPOOL BannerFile
20 PRINT "|L<BANNER>|N<USERNAME>      <USERNAME>      <USERNAME>
   <USERNAME>      <USERNAME> |M|J";
30 PRINT "<START><H>:<M>:<S> on the <DATE><ST> of <MONTHNAME>
   19<YEAR>|T|M|J";
40 *SPOOL

```

This program will generate a banner file containing the text in the example below.

3. Use a word processor, for example Wordwise or Edit. Do *not* use a WYSIWYG (What You See Is What You Get) word processor like Acornsoft View, because this generates invisible 'control' characters in the text, which will have unexpected effects on the printer.

We recommend that you do *not* use the automatic line-feed option available on most printers. If the printer always does a line feed when it receives a carriage return character, then users do not have the option of over-printing lines if they wish. In addition, it will not be possible to print files generated by **\*SPOOL** without double-spacing. (See description of **\*PRINTOUT** command in Section 3.2)

### 6.3.2.3 Example

A banner file containing this text will cause the user identifier to be printed in bold 5 times at the head of the print output, followed by the date and time at which the printing was started by the user. At the end of the user's output, a single page throw (<Ctrl-L>) will appear.

```
|L<BANNER>|N<USERNAME>      <USERNAME>      <USERNAME>      <USERNAME>
>      <USERNAME> |M|J<START><H>:<M>:<S> on the <DATE><ST> of <MONT
HNAME> 19<YEAR>|T|M|J
```

The control codes shown are suitable for an Epson printer: **lN** means 'start double sized text' and **lT** means 'start normal text'. If your printer has the auto-line-feed option turned on, omit the **lJ** (line-feed) characters from the file.

There are two banners supplied as standard: these are in a directory called **BANNERS** in the root directory \$. The file \$.**BANNERS.SIMPLE** contains the following text:

```
|L<BANNER>SJ Research File Server *** Station <STATION> (<USERNAME
>)<DATE><MTH><YEAR>  <HOURS>:<MINUTES>:<SECONDS> ***|M||J
```

which will print the banner shown as an example in Section 6.1, and a single page throw at the end of the user's output. The other banner supplied is called \$.**BANNERS.FANCY**, and contains the following text:

```
|J|J|J<MARK><USERNAME><TAB 11>***** Print started at <START><H>:<
M>:<S> <DATE><MTH><YEAR> ***** <USERNAME>|J<MARK><USERNAME><TAB
11>***** Print ended at <END><H>:<M>:<S> <DATE><MTH><YEAR>*****
** <USERNAME>|L<BANNER><MARK><USERNAME><TAB 17><USERNAME><TAB 34
><USERNAME><TAB 51><USERNAME><TAB 68><USERNAME>|J*****
***** SJ Research Printer Server *****|J<M
ARK>** Output from Station <STATION> (<USERNAME>) at <12HOURS>:<MI
NUTES> <AM> on <DATE><ST> <MONTHNAME> 19<YEAR><TAB 77>**|J*****
*****|J
```

This will print a banner of the form

```
FRED          FRED          FRED          FRED          FRED

***** SJ Research Printer Server *****
** Output from Station 068 (FRED) at 3:46 PM ON 2nd August 1986 **
*****
```

user's text

```
FRED          ***** Print started at 15:46:27  22aug85 *****  FRED
FRED          ***** Print ended at   15:48:17  22aug85 *****  FRED
```

followed by a page throw.

## 6.4 Remote despooler (optional software)

As previously mentioned in this section it is possible for print jobs to be held in the print queue by setting **printing enabled** to **No**. This allows a BBC microcomputer, running the remote despooler software, to take the data and print it out for itself. This has two advantages, first the software allows a station from a different room to print the output, second it allows more than two spooling printers on the network increasing the throughput.

For the remote despooler to share jobs with the despooler in the fileserver the **Printer exists** option must be set to **Yes**. This will allow the fileserver to despool the contents of its print queue as normal, but if more than one item is held in the print queue then the remote despooler will print it.

The remote despooler software allows a user to specify which names to search for and which type of printer to send to. The program prompts for a logical printer name and then the type of printer. For the type of printer the program expects a number in the range 0 to 4 corresponding to the value given to **\*FX 5,n**.

Despooler software below version 0.10 does not support despooling of more than one logical printer and will only print to the currently selected printer.



# Chapter 7: Utility Mode

---

## 7.1 Introduction

Utility mode is used to initialise discs, take backups and change some system parameters. Communication with the MDFS in this mode uses the program \*FAST which can run as a sideways ROM, or in RAM. With FAST running, the BBC microcomputer is acting as a *terminal* connected (via the Econet) to the MDFS. The BBC microcomputer sends characters from its keyboard to the MDFS; characters sent from the MDFS are received and printed on the screen. The actual controlling software (i.e. the bit that decides what to do when a given key is pressed) runs in the MDFS and the program is part of the File Server program (see section 4.4). The BBC microcomputer is acting as a rather elaborate Input/Output device.

## 7.2 Entering Utility Mode on the MDFS

In general, a slowly flashing Disc Free light means that either the File Server program has not been found, or that the File Server is waiting for you to press the Release Discs button. A quickly flashing Disc Free light means that the File Server is searching for the File Server program. A flashing Utility Mode light means that the Utility program has been successfully loaded and is waiting for a connection from \*FAST or a serial terminal. A steady Utility Mode light means that connection has been established.

### 7.2.1 For users with a FAST ROM fitted in a BBC Microcomputer.

From power-off, turn the File Server on with the front panel key-switch in the *System* position, make sure there is a disc with the File Server program in it and press the Release Discs button.

If the File Server is already on line, turn the key to the System position, log-on as a System-privileged user and type **\*FINISH**. If there is a disc containing the File Server program already in the system the Utility Mode program will be loaded and the Utility Mode light will flash. With no such disc, the Discs Free light will eventually flash and you should insert such a disc, press the Release Discs Button, and wait for the Utility Mode light to flash.

Now use the BBC Microcomputer, (which is preferably near to the MDFS unit to save walking,) and type **\*FAST**. Wait for the prompt:

Station number to attach to:

and then type in the station number of the File Server, which will be 254 unless you have explicitly changed it. (Changing the File Server station number is described below.)

### 7.2.2 For users with no FAST ROM.

The procedure is much the same as above except that the \*FAST program has to be loaded from the MDFS. As a system privileged user, turn the key to the System position and type **\*FAST** on a BBC Microcomputer, and at the prompt

Station number to attach to:

DO NOT enter the station number but instead type **\*FINISH**. Wait until the Utility Mode light flashes and then enter the station number.

If the message Key locked or Insufficient privilege appears after typing \*FINISH, type **SHIFT-f1**, which should produce the message **OS command: \***. If it doesn't, press **BREAK**. Either way, you will need to type **\*FAST** again, rectify the cause of the error above and re-type **\*FINISH**. Then type in

the station number as per the above instructions.

The BBC Microcomputer is now acting as a terminal to the MDFS, and the *Main Menu* will be displayed:

```
MDFS Utility Program ver 1.03
(ROM version 1.00)

A - Alter configuration parameters
B - Boot Fileserver
C - Copy whole disc
D - Add to Winchester defect list
F - Format new disc
L - List discs
P - clear Password file
R - Rename disc
S - Set File Server station number
T - Tape Menu
V - Verify Disc
Z - Park disc heads
```

Or press the front panel button  
to start the fileserver.

Command (H for help) ?

## 7.3 Using Utility Mode

### 7.3.1 General Notes

N.B The <Escape> key may be pressed at any stage to abort an operation and return to the main menu. Pressing <Return> when asked for a parameter will generally leave the previous value unchanged. For an explanation of the drive lettering conventions see section 7.5.

### 7.3.2 A - Alter parameters

This option sets some stored parameters which are held in the Battery-backed RAM in the MDFS. They control the operation of the serial printer and the floppy disc drives. The program will display the following menu:

#### ALTER PARAMETERS

Any changes will take effect  
when the file server is next  
booted except for the step rate  
which will change immediately.

Typing <RETURN> will leave the  
current setting unchanged.

Serial (RS232) parameters:

```
Baud rate :
8 - 19200 baud
7 - 9600
6 - 4800
5 - 2400
4 - 1200
3 - 300
```

2 - 150  
1 - 75  
Current setting: 7  
Option : (1 - 8)?

Key in a number between 1 and 8 to change the Baud rate to suit the serial printer connected to the MDFS. The system will then ask for the number of data bits per character, whether parity and stop bits are required to be sent, and what sort of handshake mode is appropriate. The most common values are shown here, and are the ones selected when the station number is reset (as described in §7.4).

Current setting: 8  
Number of data bits (5..8) ?

Parity:  
0 - No parity  
1 - Odd parity  
2 - Even parity  
Current setting: 0  
Option (0..2) ?

Stop bits:  
1 - 1.0 Stop Bits  
2 - 1.5 Stop Bits  
3 - 2.0 Stop Bits  
Current setting: 3  
Option (1..3) ?

Handshake mode:  
0 - CTS/RTS  
1 - Xon/Xoff  
2 - None  
Current setting: 0  
Option (0..2) ?

The final parameter set by this command is the step rate for the floppy discs. If the step rate is set too fast for a drive, the system will revert to the slowest speed. The options available are:

Floppy disc drive parameters:

Step rate:  
0 - 3ms  
1 - 6ms  
2 - 10ms  
3 - 12ms  
4 - 15ms  
5 - 20ms  
6 - 30 ms  
Current setting: 0  
Option (0..6) ?

### 7.3.3 B - Boot File Server

This will boot the File Server from Utility Mode into normal operation (i.e. On line). A File Server program (the file \$.FS) will be needed on one of the MDFS discs. Pressing the RELEASE DISCS button in Utility Mode will also attempt to boot the File Server.

### 7.3.4 C - Whole Disc Copy

This copies one disc (floppy or hard disc) entirely to another, including the entire directory structure,

password file, disc name, Account balances and Printer information. It will also format the destination drive first, if requested.

Example of a floppy disc backup:

#### WHOLE DISC COPY

This option makes an identical copy of a whole disc. Previous contents of the new disc are destroyed, including disc name, account information etc. The new disc need not have already been formatted.

Source drive (A..H) ? **A**  
Destination drive (A..H) ? **B**  
Format destination disc (Y/N) ? **No**

Put source disc in A & push space **<space>**  
Name = floppy1 size = 800K

Put destination disc in B & push space **<space>**  
Name = olddisc size = 800K

Copy complete - Verifying...  
Verify complete - No errors

If the destination disc is to be formatted, the name will not be given as it will probably not be defined, so you must be careful that you are formatting the right disc. Formatting a floppy disc will take about two minutes and so will copying a whole disc.

The File Server memory cannot hold the contents of a whole floppy disc; so if the source and destination drives are the same, the system will prompt for you to insert the discs alternately several times.

It is also possible to copy between two winchesters using this command. This takes about 20 minutes.

### 7.3.5 D - Add defect list

This command is used to tell the winchester controller about bad sectors on a winchester. It is not supported by all makes of controller (notably the ACB 4000A and ACB 4070), but is supported by the RO752 drive.

If you have any bad sectors (and no drive is guaranteed wholly error-free) you will have probably noticed them from messages printed out while the File Server was On line. You should record the error number and block number of all such errors (so that you can see if the block is the same etc.). As soon as you have discovered that you have errors, you should verify the disc probably two or three times. Some errors are termed *soft*; these are sectors that can sometimes be read and sometimes not. Sometimes there is a power glitch which means that a sector could have been written badly (as opposed to the disc surface having a flaw on it). N.B. The block number printed by the File Server will be a *logical* block number, this will not be the same as the physical number printed by the verify command. It is the physical block that must be entered into the Add Defect command. Entering a number that does not actually have an error in it will cause the message Sector read OK 5 times - are you sure? Discs with more than about six errors should be regarded as suspicious and may require re-formatting.

### 7.3.6 F - Format New Disc

This option will format floppy discs or hard discs and then write the necessary header information onto the disc. Any previous data on the disc will be destroyed. There is no need to format floppy discs before using the C (copy) option, as this option can format discs for itself.

For example:

## FORMAT NEW DISC

Format disc in which drive (A..H) ? **B**  
Disc name (max 10 chars) ? **green<return>**  
Push space to format disc in drive B<space>  
Formatting...  
Format complete  
Writing new root  
Verifying...  
Verify finished -  
All sectors OK

Format another disc (Y/N) ?

Now an example of formatting a hard disc with an ADAPTEC ACB 4000A controller, such as are fitted to most BBC-compatible Winchester disc drives. Please note that option 'B' should be tried first, even if your drive did not come from Acorn. This option assumes that the disc has already been formatted (but probably for another machine).

N.B. New defects can only be entered at format-time, due to limitations in the Adaptec controller (with a Rodime RO752, as supplied by SJ Research, defects can be entered at any time without having to re-format the disc).

Format disc in which drive (A..H) ? **E**

That is a Winchester disc  
- are you sure (Y/N) ? **Yes**

What sort of drive is it:

- A - SJ (Rodime RO752)
- B - Acorn (Adaptec 4000A)
- D - 20Mb Half height NEC/Mitsu
- E - 40Mb Half height NEC/Mitsu
- F - 20Mb Full height NEC/Mitsu
- Z - User defined drive

Enter option: **B**  
Enter defect list (Y/N) ? **Yes**  
Defects must be entered in ASCENDING  
order (maximum 16)  
All numbers should be in decimal.  
Cylinder: **4**  
Head: **2**  
Bytes from Index: **1002**  
Another defect (Y/N) ? **No**  
Disc name (max 10 chars) ? **HARD-0<return>**  
Push space to format disc in drive E  
Formatting...  
Format complete  
Disc size = 31200k

N.B. The disc size printed may not correspond exactly to the notional drive size. This is due to several factors. Firstly the MDFS uses a sector size of 512 bytes which increases the storage capacity of the drive. Secondly it rounds the disc size down to the nearest multiple of 5200k, and thirdly there is a maximum partition size of 36400k (until multiple partitions are supported this is the maximum size of the usable part of any one drive; multiple partitions can be added at a later stage). This 36400k limit is due to the amount of data that you can store on a tape drive.

Verifying...  
Verify finished -  
All sectors OK

### 7.3.7 L - List Discs

Tries to read all the discs (A..H), and prints information on any that are connected to the File Server. It takes about a second to test for each disc, so do not be suprised if nothing happens immediately. For example:

#### LIST DISCS

Discs currently available:

A:	Name: Floppy1	size: 800K
B:	Name: green	size: 800K
E:	Name: Main	size: 20800K

### 7.3.8 P - Clear Password File

This option is required if the password file has been corrupted on a disc, for example by saving rubbish over it, or if there exists no disc with a system user on it. (In general it is possible to work on the password file on any disc, by inserting another disc which has in its password file a system privileged user, with access to account 0)

The prompt will be:

#### CLEAR PASSWORD FILE

This option will remove a corrupt or unwanted password file from a disc.  
The disc will be left with  
NO PASSWORD FILE AT ALL : for security  
a new password file should be written  
(using EDITPASS) as soon as possible.

Clear disc in which drive (A..D) ? **A**  
Insert disc & push space when ready  
Name = floppy1 size = 800K

OK (Y/N) ? **Yes**

If a disc without a password file is installed into the system, **anyone attempting to log on will have system privilege**. For this reason, it is advisable to use EDITPASS as soon as possible, to create at least a null password file on the disc.

The **F** (format new disc) option creates a null password file automatically.

### 7.3.9 R - Rename disc.

All File Server format discs have a name associated with them. The name is up to 10 characters long and is subject to the usual rules governing filenames. When the MDFS is On line the only way for a client to select a particular disc is by its name, so it is wise to give each disc a unique name. In particular, the **C** option copies the name from the source disc, so the destination disc should be renamed if it is to be used for reading data off.

The process is:

#### RENAME DISC

Rename disc in which drive (A..F) ? **F**  
Name = blue2 size = 20800K  
New name (max 10 chars) ? **hard1 <return>**

OK (Y/N) ? **Yes**

### 7.3.10 S - Set Station Number

This changes the station number of the MDFS. This number is stored in a special memory that is maintained by an internal rechargeable battery. No maintenance is necessary, but if the MDFS has been out of use for more than about 6 months, the battery may have run down. In this case, follow the instructions in §7.4 to reset the station number to 254, then change the number (if required) with this option.

Any change in station number will be implemented when the File Server is next booted. The prompt will be:

```
Current station number is : 254
New station number       : 253 <return>
```

### 7.3.11 T - Tape Menu

This allows tape backups, etc to be made. For information about this menu see Chapter 8.

### 7.3.12 V - Verify Disc.

This reads all sectors of the specified disc. It tells you about any sectors which are unreadable or which are imperfect but readable (these are referred to as 'dodgy'). If a disc has any bad sectors on it you should try verifying the same disc in a different drive. We suggest that you cease using that disc and transfer all data to another disc, preferably using MULTICOPY. The same applies to a disc with dodgy sectors if the data on that disc is valuable.

### 7.3.13 Z - Park disc heads

Winchester disc heads never actually touch the disc itself while in operation, but on most drives they do rest on the disc when the unit is switched off. It is usual therefore, to move the heads to an area of the disc which is not used for data storage before the power is removed, and especially before transit. Some (older) drives even have a screw to secure the heads: refer to your drive manual for information. The Z command moves the heads to the innermost track of the disc (and beyond sometimes). This has the same effect as pressing the RELEASE DISCS button while the File Server is on line. N.B. Adaptec controllers take a long time (upto 10 Seconds) to reload the heads when the drive is next accessed. Any such access will automatically unpark the heads.

## 7.4 Re-setting the Station Number to 254

If the RELEASE DISCS button is held pressed while the power is turned on then the File Server station number will be set to 254 (the normal default). This will normally be necessary only in two circumstances:

The unusual circumstance of the File Server number having been forgotten (although it could often be found by running \*STATIONS from another File Server).

The rechargeable battery that maintains the memory storing the station number has failed, usually because the MDFS has not been powered for more than about 6 months, this will cause a flashing system failure led. In this case, leave the unit switched on for about 10 minutes, switch off, and then switch on again with the button held pressed. Note that the parameters set by the A option in Utility Mode are kept in the same memory device: these will have been reset to their default values if the battery had failed. They can be changed back using the A option above.

## 7.5 Drive Letter Allocation

This section describes how the drives are labelled on the MDFS.

## 7.5.1 Physical Drives, Logical Discs and Partitions.

A complete physical drive is denoted by a single capital letter. Floppy discs are labelled A,B,C and D (as printed on the back of the MDFS), and winchester discs as E,F,G,H,I,J,K and L.

Some such winchesters may be too big for some parts of the system to cope with, and so are *partitioned* into smaller chunks. A partition of a drive is referred to by a letter followed by a number, thus E2 refers to the second partition on physical drive E. The base partition is 1, thus there is no 0th partition.

Logical disc numbers refer to a partition (but this may be the entire disc), and are only used when the File Server is On line. Such numbers are printed by \*FREE. However to select a disc using \*SDISC, you can only use disc *names* (although it would be possible to write a program to select a disc by number).

The logical disc numbers are allocated by the MDFS every time the File Server is started up or the button is pushed. The MDFS checks for the presence of drives E through L, and then checks all partitions of such drives for valid disc headers (and will print the message `Block 0 corrupt` if the SJ Research header is missing), allocating a successive numbers to each non-corrupt partition. It then does the same for the floppy drives A through D. **N.B.** This means that if you have a floppy disc in drive A and a winchester connected as drive E, logical drive 00 refers to drive E and logical drive 01 refers to drive A, and not the other (more obvious) way round.

Disc Error messages contain either logical disc numbers or physical drive letters. The former is intended to be phased out as soon as possible, but was still extant in File Server code version 0.A3.

## 7.5.2 Winchester Disc Lettering

Winchester discs are connected to the SCSI bus connector. The SCSI bus talks to the disc *controller*, there being a maximum of 8 controllers (numbered 0 through 7) on any SCSI bus. Controllers 0,1,2 and 3 are allocated for disc drives, Controller 4 is for the Tape Drive, Controllers 5 and 6 are spare and Controller 7 is the MDFS itself. Some winchester disc drives have integral controllers (such as the SJ Research drive), the rest (like the Adaptec ACB 4000A controller as used by BBC-compatible winchesters) have a separate controller and drive. The Adaptec will also support two drives per controller.

### Table of Drive Letter Assignments:

(Drive 1 not supported by RO752 drives)			
E	Controller 0, Drive 0	G	Controller 0, Drive 1
F	Controller 1, Drive 0	H	Controller 1, Drive 1
I	Controller 2, Drive 0	K	Controller 2, Drive 1
J	Controller 3, Drive 0	L	Controller 3, Drive 1

Thus if you have 2 SJ (RO752) winchesters you will have drives E and F, whereas with one Adaptec controller and two drives you will have drives E and G. See Appendix C for details on how to select controller numbers and drive numbers.

# Chapter 8:

## Use of the MDFS Tape Drive

---

This chapter is split into the following sections:

Section 8.1	Introduction
Section 8.2	Using the Tape Drive in Utility Mode
Section 8.3	On Line Tape Operation

### 8.1 Introduction

The Tape Drive is used to take backups (ie copies) of any winchesters connected to your MDFS system. Most tape operations are done in *Utility Mode*, although the MDFS can be set up to take a backup in the middle of the night when nobody is using the system; in this case the MDFS will automatically go into utility mode, take a backup and re-boot.

Each tape cartridge holds approximately 39Mbytes of information on it, hence is large enough to take a complete copy of a 35Mbyte winchester. Winchester larger than 35Mbyte will automatically be partitioned into 35Mbyte chunks, hence a 50Mbyte winchester will require two tapes to take a complete copy of the disc; one a copy of the 35Mbyte partition, the other a copy of the 15Mbyte partition.

Having taken a backup, there are two ways of accessing the data stored on the tape. We can copy the entire contents of the tape back onto the winchester, and this is called *restoring from tape*. This might be used after some disastrous loss of data, due to directory corruption etc, or possibly a head crash or some other mode of winchester disc failure.

Alternatively we can restore individual files from the tape. This is done by the selecting the tape as another disc using the psuedo-directory `%TAPE`, while the MDFS is *on line*. We can then use all the usual commands like `*DIR` and `*LOAD` and the BASIC program "COPIER" to load the file from tape and transfer it back to the winchester. You might use this technique after an accidental deletion of a particular file (see section 8.3).

#### 8.1.1 The Tape Drive

The Tape Drive is a bit like a cassette player in that it has a capstan to move the tape backwards and forwards and a read/write head. The main difference is that the head in this case can move up and down across the tape into a total of 24 defined positions. These create 24 *serpentine tracks*, so called because if a particular track is read/written in one direction, the adjacent track will be recorded in the opposite direction. The tape drive is also capable of moving to a particular position along one of these tracks and this gives it a *random-access* capability which is what it uses to read a particular file from a tape.

The red LED on the front panel indicates that the tape is moving or that the head is moving up or down. Do not press the eject bar while this LED is on (not normally, anyway).

#### 8.1.2 Inserting a Cartridge and Autoload

The tape cartridge (type DC2000), when correctly inserted, is held firmly by the drive. It is not possible to insert the cartridge the wrong way round apart from the 'obviously' wrong way round (i.e. sideways on). Directly after insertion the drive will start to whirr, executing its *autoload* sequence. (N.B. If nothing happens after you insert a cartridge please check the power connections to the tape drive, and inform SJ Research.) The first part of the *autoload* sequence is a *conditioning pass*, and is a standard feature of all tape drives. The drive will wind the tape to one end of the tape and then wind it all the way back to the beginning again. After this it will continue to wind the tape back and forth while it determines exactly where the edge

of tape is, sets the read gain and a number of other parameters.

The sequence is complete when the drive has stopped whirring and the red LED has gone out. This can take anything between 1 and 3 minutes, and it is only after this that the tape can be written or read. Trying to access the tape (using the tape menu) during autoloading will give an appropriate error message.

### 8.1.3 Removing the Cartridge and *Unloading*

While it is possible to remove the tape at any time by pressing the eject bar (although preferably not while the LED is on), it is best to *unload* the tape prior to ejecting it. This winds the tape to the end, exposing only a blank section of the tape, reducing the possibility of data corruption and cutting down the time for the next *autoload*. Unlike autoloading, *unloading* is not automatically initiated by the drive and you have to initiate it by:

In *utility mode*:

Either a) Use the 'U' command

Or b) Use the option after quitting the Tape Menu

Or c) Use the auto-reboot option when taking a backup (as this also automatically unloads the tape)

When *on line* use the command \*UNLOADTAPE.

*Unloading* is complete when the red LED goes out, which takes less than a minute. You may then press the eject bar and remove the cartridge.

### 8.1.4 Tape Cartridges

Like floppy discs (and hard discs), tape cartridges require *formatting* before you can read or write to them. Of the two cartridges that come with the tape drive, one will have already had some information written onto it during the test procedure, and hence will be formatted. The other will be blank and will require formatting, using either the 'B' or 'F' options. Tapes may be reformatted if you wish (but this is not usually necessary).

We can also *write-protect* the tape by moving the little black tab against the direction of the arrow. This can be used to protect against accidental mistakes on the part of the user. But bear in mind that if you do decide to write to the tape you will have to *unload* the tape, move the tab and reinsert the tape; this may take several minutes.

### 8.1.5 Tape Cartridge Life-expectancy

Tape Cartridges have a limited lifespan measured in *passes*. Winding the tape from the beginning to the end is considered to be one *pass*. Thus an Autoload sequence does two passes. The DC2000 Cartridge is specified for up to 5000 passes, and an attempt to record the number of passes is made, the information being written onto the tape itself. This number (which is read by the 'P' option) represents the number of passes over the tape during any Format or Backup operations. Since backups will probably represent by far the greater proportion of tape use this should be a good indicator of tape usage.

### 8.1.6 Managing Tape Backups

The two cartridges supplied with the system provide the bare minimum for taking backups (with only 1 cartridge, while the backup was taking place there would be no consistent copies of the disc, and if the cartridge itself failed there would be no older backup to rely on, nor any means of testing the system). With two tapes backups should be taken *in rotation*; time-stamping of the tapes means that you can easily identify which is the younger of the two. A *rotational* system can be explained as follows: suppose you have three tapes called 'A', 'B' and 'C'. If you used tape 'B' for your last backup then use 'C' for the next, then 'A' and then use 'B' again for the one after that.

The particular backup system (i.e. number of tapes, frequency and method of backup) depends on what security you want, how important your data is and what effect there would be if a disc did go down, etc. If your system has two discs then one could decide to backup one of them more frequently than the other, having one disc as a fairly *static* disc, and the other as a main *work* disc.

In deciding what security is needed, it is worthwhile to examine the possible causes of data loss; these fall into seven groups: human error, mains power failure, File Server software bugs, File Server hardware failure, Disc failure, Drive failure and possibly malicious users, if you are unlucky.

Human errors will occur the most frequently; accidental deletion (and use of wildcards without due caution), over-saving with the wrong program, over-saving with a 2-byte file (i.e. forgetting to type OLD after pressing BREAK), removing a disc without pressing the Release Disc button (applies to floppies only). They usually only affect a few files, and these can be read back individually from a backup tape.

The effects of drive failure are usually spontaneous and total. That's right, no warning and you've lost the lot. And yes it happened to us in May 1986. But luckily we had a backup tape, replaced the disc with a new one and restored the data.

The other categories can have effects anywhere from corrupting the name of a file to wiping out large chunks of the directory structure. Another big advantage of frequent backups is that it means frequent re-booting of the fileserver, during which the directory structure is checked for inconsistencies, identifying the 'dormant' bugs.

Backing up every evening is a good method to use, and if your data is valuable (e.g. you have some pupils doing exam work on the system) then IT IS ESSENTIAL. The problem with backups daily is that you don't get a lot of history, so that if someone has accidentally deleted a program or it has got corrupted a week ago, you probably do not have a copy of it on a tape. To get round this problem you can, in addition to a daily backup scheme, have a weekly backup scheme, also using a rotating sequence of tapes. This will give you a few weeks of history as well as an up-to-date copy to guard against major disasters.

## 8.2 Using the Tape Drive in Utility Mode

All tape operations must be done from the *Tape Menu*. Press 'T' from the *Main Menu* to get this. You can get out of the Tape Menu by pressing 'Q' or ESCAPE. In normal use you will probably only use the 'B' and 'P' options from the menu.

### TAPE MENU

```
B - Backup winchester to tape
C - Check winchester against tape
F - Format tape
N - Name tape
P - Print tape information
Q - Quit tape menu
R - Restore winchester from tape
U - Unload tape
V - Verify tape
```

Tape command (H for Help) ?

### 8.2.1 Using the *Backup* command

This option allows you to do more than just take backups, like formatting, re-booting the MDFS, unloading the tape, and waiting for the autoloader sequence to finish, so you'll probably use it most of the time.

#### Example Backup

Here is an example for an MDFS with a single 20Mbyte disc drive, and a formatted tape in the tape drive already autoloader. Total time for backup is about 30 minutes.

Tape command (H for Help) ?**B**

BACKUP WINCHESTER TO TAPE

Winchester discs available :-

E: Name: BLANK-DISC size: 20800K

Choose disc to backup from (E..H)? **E**

Boot FS after backup (Y/N)? **No**

About to write to tape:

Name: A-Tape

Descriptor:

Everything OK (Y/N)? **Yes**

Copying...

At this point the MDFS is now reading data from the disc and writing it to the tape drive. The tape drive LED should be on continuously and the disc LED should flash a few times a second. This part will take about 15 minutes for a 20Mbyte disc.

Checking...

The MDFS is now reading data off both the disc and the tape drive, comparing the data bytes as they come back. Again this phase will take about 15 minutes.

Backup completed OK.

### More General use of the Backup command

After you have chosen which disc you wish to backup, the MDFS will read the tape cartridge. There are 4 possibilities:

- a) The Tape is formatted and the ID sector OK - as in the example above.
- b) There is no tape in the drive or it is autoloading. The MDFS will print

Wait for tape (Y/N) ?

You would normally press 'Y' here; this will cause the MDFS to wait for the tape to finish autoloading just prior to taking the backup proper.

- c) The tape is unformatted. The MDFS will print

Tape is unformatted  
New tape name (max 10 chars) ? **Tape4**

- d) The tape is formatted but the ID sector is corrupt. Very unlikely. The backup operation will be aborted.

The next question that the MDFS will ask is whether you wish to reboot the FS after taking the backup. If you enable this option, the MDFS will also unload the tape afterwards. If the backup subsequently fails, then the FS will not reboot the system to alert the system manager to this fact.

### Errors during the Backup command

During the *Copying* and *Checking* phases of the backup, most errors will not cause the backup to abort. Because the MDFS is read/writing to both the winchester and the tape, errors will either be tape errors, disc errors or *data corruption* errors.

Disc errors should never cause the backup to abort except if there are too many of them (more than about 20) in which case you will get the message TOO MANY ERRORS. As far as tape errors go, error numbers 10 to 18 are not considered fatal, but any other error numbers will abort. Errors in the latter category are usually permanent, i.e. they won't go away without human intervention (such as inserting a different cartridge). Errors in the former category are dealt with after each of the *Copying* and *Checking* phases. Such bad sectors as have been encountered are *reassigned*, the data is rewritten to an *alternate* (i.e. 'spare') sector and the sector number is entered into the *bad block list*, which is also written back to the tape. Problems can occur when this list becomes full, and if this happens the tape must be reformatted which will provide a new set of 'spare' sectors.

The system maintains a *flag* (i.e. a piece of information) on the tape as to whether the the backup succeeded or not. Before the backup occurs, it writes this flag as meaning 'Backup not OK'. After it has written the whole tape, read it all back again and verified that all the bytes were correct it then goes back and rewrites the flag to mean 'Backup OK'. However not all errors encountered on the tape will cause the backup to fail; the MDFS will attempt to recover from most error conditions. If this recovery procedure is found to work correctly, then the flag will also be set to 'OK'.

### Aborting Backups

You can if you wish interrupt the backup process, by pressing the ESCAPE button. With reference to the above paragraph, this will leave the *flag* mentioned in the 'Backup not OK' state. Note that this (in this particular circumstance) will not return you to the main menu, but leaves you in the tape menu.

### Taking a Backup in the evening, example:

- a) Get the MDFS into Utility Mode
- b) Insert a tape cartridge
- c) Select the tape menu
- d) Press 'B'
- e) Select the disc
- f) Reply 'Y' to Wait for tape?
- g) Reply 'Y' to Boot FS after Backup?
- h) Reply 'Y' to Everything OK?
- i) Go home and let the MDFS do the rest.

### 8.2.2 Formatting tapes

As mentioned earlier, you can also use the Backup option to format tapes. During format, the entire tape is rewritten, and then read back again to determine whether there are any bad areas of the tape. These are entered into a list which is then stored on the tape. If the tape is reformatted, all areas of the tape that have subsequently found to be bad (during the backup process) will be considered bad automatically. The overall tape size is set after each format operation.

Tape command (H for Help)? **F**

FORMAT TAPE

Formatting will take about 40 minutes.

New tape name: **B-tape**

Are you sure you want  
to format (Y/N)? **Yes**

Formatting...

Format complete:

Bad blocks -

Primary: 34

Growing: 0

Tape Size: 38664k

The number of *Primary* bad blocks is the number of bad sectors found during the format operation and includes all the bad blocks found during previous format and backup operations. The larger this number, the smaller the tape size. More than about 40 bad blocks on a freshly formatted tape should be considered excessive. The number of *Growing* bad blocks is the number of bad sectors found during a backup operation.

### 8.2.3 Getting tape information

Tape command (H for Help)? **P**

PRINT TAPE INFORMATION

Name: DiscE--2

Descriptor: Rotating Backup

Bad blocks -

Primary: 34

Growing: 0

Tape Size: 38680k

Number of tape passes: 154

(Cartridge is spec'ed to a max of 5000)

Last formatted: 05:54 09nov86

List of discs backed-up:

Name	Size	Time	Date	State
pupil-DISC	20Mb	16:17	29dec86	OK

### 8.2.4 Restoring data from a tape backup

Tape command (H for Help)? **R**

RESTORE WINCHESTER FROM TAPE

List of discs backed-up:

Name	Size	Time	Date	State
Teacher	20Mb	16:17	29dec86	OK

Winchester discs available :-

E: Name: BLANK-DISC size: 20800K

Choose disc to overwrite (E..H)? **E**

Everything OK (Y/N)? **Yes**

The restore operation will take 15 minutes for a 20Mb disc.

N.B. Restoring from a tape will completely overwrite the entire disc, including the password file, printer setup (as in Editprint) and File Server software. The latter may cause the system to run an entire version of the File Server code. Please check that your version number (using \*VERS) and copy on the latest version if necessary (see section 4.4).

## 8.2.5 Other operations

### Unload tape

Try to get into the habit of using this command. Although it is not essential that you do this before removing the tape, it is wise because it leaves an unsensitive part of the tape near the window. This command will return a prompt before the drive starts to whirr - please wait until the red LED goes out before you remove the tape; it takes about 30 seconds to unload the tape.

### Check tape against disc

This command reads the disc and the tape, comparing the data read byte-for-byte. In fact the Backup command does this operation anyway, and if you reboot the filesaver after a Backup and then do a Check, you will get the error **Data fail @ 8** (but you shouldn't get any errors after that). This is because the filesaver will have written the current time to all discs in the system as soon as it has gone *on line*, making the data on the disc different to that on the tape.

### Verify tape

This operation reads each sector, making sure that it is readable, but ignores the actual data bytes read back. In version 1.03 it only verifies the area used by the last backup, and takes about 15 minutes for a 20Mbyte verify.

### Name tape

Allows you to change the name and/or descriptor of the tape. The *name* is a 10 character (max) name, with character restrictions, whilst the *descriptor* can be up to 80 characters long, with any characters, most importantly spaces, allowed.

## 8.3 On Line Tape Operation

When the tape is accessed with the MDFS on line, the tape is seen as a (rather slow) read-only disc; the data on this 'disc' is the data written onto it during the last backup. A tape cartridge can be inserted at any time, and once it has finished autoloading, you can use all the usual commands \*DIR, \*CAT, LOAD etc to move around the tape directory structure and read a particular file from it. You cannot use any commands which would write to the media, e.g. \*ACCESS, \*RENAME, \*DELETE, SAVE etc. Access to files on the tape is subject to the usual accounting rules, just like an ordinary disc. The root, \$, of the tape is the pseudo-directory %TAPE, so to catalogue the root type **\*CAT %TAPE**. If you select the tape as your current directory, to select an ordinary disc you will need to refer to its discname, or you can use **\*DIR<return>** or the character & (ampersand).

The main point to remember is that because the tape is slow it can often take a very long time to retrieve a directory or file, long enough to cause the BBC microcomputer to time-out and produce the error message No reply. The programs COPIER and MULTICOPY take advantage of the ability of ANFS to increase this time-out value to several minutes, so you will find it easier to use machines with ANFS if you have any (BBC microcomputers have NFS as standard).

The command \*DIR has particularly nasty side-effects if it generates a No reply error as any subsequent communication with the File Server will result in a Channel error. The only way out is to log-on to the File Server again. The caching on the File Server comes into its own when dealing with tapes, and subsequent access to data will be almost instantaneous (provided of course that the system is not being heavily used by other clients causing tape data to be thrown out to make room for new disc data). If you want to use \*DIR with safety, therefore, type **\*INFO <dir.>** to cache the data, wait for the tape to stop winding, and then do the \*DIR.

The error %TAPE not found is generated by a reference to %TAPE with no tape drive connected, no tape cartridge in the drive, or before the drive has finished autoloading. When you have finished using a cartridge you should use the command \*UNLOADTAPE, which will initiate the drive's unload sequence. When the red drive LED has gone out you may then remove the cartridge. If you remove the tape without issuing \*UNLOADTAPE. You will be able to access that part of the tape data which is already in cache, but as soon as a new sector is required from the tape the File Server will sense that the tape has gone and will report Drive error. The disadvantage is that the tape will be left with a sensitive area of the tape in a rather exposed position.

### 8.3.1 Example File Recovery

This example is applicable in the case where the full pathname of the file in question is known and a BBC microcomputer is being used (i.e. no ANFS):

You will need to repeatedly load the file from tape until it is fully cached. A function key is therefore a useful method. To recover a BASIC program, for example:

Type: **\*KEY 0 LOAD "%TAPE.FORM1.Jim.Work.Addresprog"|m**

and for any other sort of file (less than 31K long),

Type: **\*KEY 0 \*LOAD %TAPE.Fiona.Project.teletxt001 8000|m**

and then press function-key 0. The tape will start to wind while it is searching through the pathname. After about 30 seconds, if the tape is still winding, the BBC microcomputer will give No reply. Wait until the tape stops winding (the File Server will lock your station out anyway until the tape has finished), and then press function-key 0 again. Repeat until the LOAD completes without error. It will do this eventually as gradually more and more data will be in cache, reducing the amount of time taken to transfer the file. At this point you may save the file directly (if it was a BASIC program), or type **CHAIN"COPIER"** and transfer the file again.

On a machine with ANFS, use COPIER directly:

**CHAIN"COPIER"<return>**

Source Filing System **\*DIR %TAPE<return>**

Dest. Filing System **\*DIR :<discname>**

File name:**form1.jim.datafiles.addresses<return>**

New name:**<return>** (uses the same filename)

File name:**<escape>**

N.B You will now have %TAPE as your current directory, so remember to select a directory on another disc before you do anything else.

# Chapter 9: Dealing with problems

---

## 9.1 Overview

The important thing to remember is that computers do crash. Even assuming no software bugs and perfectly working hardware, accidents can and will still happen. Power cuts are a major cause of problems, with lightning a close second.

The first rule to remember is **take frequent back-ups**. This applies especially to Econet File Servers, where the users themselves have absolutely no control over their file integrity. In many cases a File Server may be your only mass storage device. Sooner or later **IT WILL FAIL**. The only guarantee against disaster is to take a back-up every day.

Reliability of the File Server can be improved in other ways too. The ideal environment is a firm shelf in a well ventilated position, out of direct sunlight, where people are unable to shake the machine. If possible the File Server should be left switched on at all times. Winchester discs are most likely to be damaged during the critical power-on sequence, when the heads are not flying in the normal way but are rubbing on the surface of the disc.

## 9.2 What to do if the File Server crashes

The most likely symptom is all the computers attempting to access the File Server return the error **Not listening** or **Station 254 not present**. If *any* machines are still communicating with the File Server then it is a not a problem with the File Server.

If the performance of the machine varies on its position in the network then suspect the clock speed. A standard problem is that a BBC machine has problems loading a large file, this is because the clock is running too fast or too slow. Check that both terminators are plugged into the far ends of the network. Similar problems may occur if the network has been installed with long spurs : see section 9.4 for debugging a correctly installed network, see the clock manual for installing a network for the first time.

Press the release disc button, if this has no observable effect then turn the MDfS off at the keyswitch. Restart the File Server, make a backup copy of the File Server discs then telephone SJ Research to report the problem.

### 9.2.1 Mains Electricity

It is always worthwhile checking mains installation if you are getting computers crashing or line-drivers blowing. A simple and cheap tester can be obtained from RS components, part number 424-709. This device can be used to test all sockets, including mains extension leads.

## 9.3 Network security

This section deals with the problems of protecting data on Econet systems generally and on SJ Research File Servers in particular.

There are two separate areas to be considered, accidental and deliberate interference. Accidental problems can be further subdivided into *mistakes* by the users such as saving one program over another, or deleting too many files with an injudicious wildcard; and *external accidents* such as turning off the File Server at the wrong time or a lightning strike. Deliberate interference also divides into two categories: *snooping* or passive activities to gain access to other people's data, and *wrecking* or active alteration of the data.

### 9.3.1 Accidents

No-one can guard against mistakes completely but users can easily be protected from many of the more

common errors.

The most common ways that beginners lose files are:

1. Saving over a file with a new program of the same name.
2. Saving a *null* program (in the case of Basic this will be two bytes long) because they have not typed **OLD** after pressing <Break>.
3. Using wildcards in a delete statement with more effect than intended.

Encourage users to use names which are meaningful. Ten characters is enough to choose a sensible name, especially if the use of hyphens and underlines is encouraged.

Suggest that users check their saves with a **\*INFO <filename>** after each save, and explain how to check the file length. Alternatively use **\*OPT 1,1** to display this information automatically.

Beginners can be given a lot of protection by initially giving their directory a default access which is locked (for example **\*DEFACCESS LWR/**). The disadvantage of this is that it is then necessary to type **\*ACCESS <filename> -L** before a new version can be saved over the existing one. If you plan to use programs like EDWORD which do not allow **\*** commands then this is a disaster!

Beginners may also appreciate the **No Short Saves** option which can be set by the system manager using the EDITPASS program. There is also an option to require the use of **\*ENABLE** before any wild card delete operation is permitted.

### 9.3.2 Recovering from Mistakes

*Lost* files can be recovered fairly readily if the system manager has been careful about taking regular backup discs (using the system in Utility Mode). It is recommended that copies of all current discs are taken weekly (more often if there is critical data), and stored in a safe, preferably locked, place. It then becomes relatively simple to insert the backup disc, recover the lost file from it, then remove and put away the backup again.

Backup discs may be used in rotation, but there is a lot to be said for keeping long term archive discs as well. The reason for this is that it may not become apparent that a file has been lost or corrupted until some considerable time has elapsed, and that all the backups may have been rotated through the system by then.

### 9.3.3 Deliberate Interference

#### Care When Logging On

One of the most common security hazards is using an unprotected machine. Before doing anything sensitive, remember to run **\*PROT** on your computer. In particular, this is worth doing **BEFORE** logging on as a system privileged user. If you do not do this, your memory can be examined by any other machine with the appropriate software. (Note that this problem is not specific to SJ Research File Servers, and indeed remains a problem even if there is no File Server on the network at all.) Protected status is cleared by <Ctrl-Break>.

#### Remembering to Log Off After a Session

The File Server and your local station both store information about you. Pressing <Ctrl-Break> or even switching off your local machine will not affect the File Server's status, so that only a small amount of trial and error will allow another user to re-enter the system from the same terminal (you can show this to be so by switching off your own terminal and typing **\*USERS'** from another machine.). It is essential for any user who is concerned about security to type **\*BYE** before leaving his terminal.

It is also good practice to turn off your machine after a session, in case any secret information has been left lying around in RAM (the EDITPASS program is a good example of this).

## Passwords

If a user gains access to your files in a *legal* way, there is very little that you can do about it (but see the next section for one way of limiting the damage.)

The difficulty in ascertaining someone's password by trial and error increases rapidly with the number of letters in the password. Remember too that numbers and other non-alphabetic keys (!, -, \_, .) etc) can be used. The password is governed by the same restrictions as for filenames. All passwords should be at least five letters long, preferably more. The maximum length is ten characters. Users should not use passwords which might be guessed easily by others (do not use your wife's name, phone number or car number, nor do we recommend the use of characters from Tolkien, 2000AD or Hitch-hiker's !).

Take care that you do not leave your system unprotected whilst you are still learning it: it is very hard to re-establish security once it has been broken. One enterprising hacker managed to add a command to EDITPASS, to save a copy of the password file into his own directory. By placing a <ctrl-U> character before and a <ctrl-F> character after his code, he ensured that it would not be listed on the screen of a BBC Micro.

## Keys and Key Discs

The Modular Disc File Server requires the key switch on its front panel to be turned to the SYST position before any system privileged operations are permitted. Make use of this feature all the time, ensuring that you do not leave the key in the SYST position if you do not require system operations.

The File Server can be further protected from unauthorised use by having only one (or a few) discs on which system privileged users exist in the password file. These discs can be kept locked up except when system operations are required. Before performing a system privileged operation press the RELEASE DISCS button and change one of the discs for that containing the system user(s).

## Snooping and Wrecking

Provided that passwords do not fall into the wrong hands, the opportunities for deliberate meddling can be fairly limited. Obviously it is useful to make all sensitive areas of the disc **Private**. Non-private directories can also be made reasonably secure by setting the access to **WR/**. Note that if the Default access includes public access, there will always be a period during the creation of a new file when other people can read it, even if it is then rapidly set to **WR/**.

## Care when not using the network

Remember that the network is still active, even if you are not using it. For example if you are editing files to a local disc, your screen can still be viewed ! This could have serious consequences if you were writing exam papers, for example. Use the program \*PROT (preferably in your boot file where it will be run every time you log on) to avoid all but the most dedicated hacker. The 100% safe way to avoid hacking is to unplug your system from the network when doing work of a particularly sensitive nature.

## 9.3.4 Special Techniques

### Limiting the Opportunity for Damage

Editing the password file may only take place when the File Server key-switch is in the system position. This editing allows a system privileged key-holder to remove everyone's access to a particular account (including his own access) and then *lock* the password file. In this state, no-one may access private files that have been saved in that account, not even the key-holder. The password file will have to be re-edited to enable access again. This technique may have some use for files which are frequently read but only rarely changed (e.g. libraries or time-table information), which may be made **WR/R** and saved into an account which is not normally accessible. Take care not to remove your own access (as system manager) to account 0, otherwise you will not be able to edit the password file.

## Restricting read access to files and restricting write access more.

General Principle: Allocate two accounts, one for read access only and one for full read-write access.

Create a private directory in the Write account, with auxiliary access in the Read account and Defaccess WR/R. Only those users with access to one or other (or both) of the two accounts will be able to see or enter this directory.

When a new file is created within this directory, set the auxiliary account to the write account, thereby removing from the read account the owner access to the files, giving them intermediate, *read-only* access rights.

Example: Pupil Reports are required to be available for inspection by any member of staff, but only the master in charge should be able to alter them. The master in charge should have accounts 2 and 3, the other staff account 3, and the rest of the users access to neither account.

### Method:

1. The creator selects the place where he wishes the new directory to appear.
2. \*CDIR reports
3. \*ACCOUNT reports 02 (03)
4. \*ACCESS reports +P
5. \*DIR reports
6. \*DEFACCESS WR/R

The directory is now ready for use. When saving a new file, type:

7. \*SAVE <file name> 0+0
8. \*ACCOUNT <file name> (2)

then save the file you want as <file name>. This avoids a malicious user writing to the file between it being saved and its access being changed.

## 9.4 Debugging the network hardware

This Section contains a guide to finding faults on an Econet system.

If you are having problems which result in no communication at all, or only unreliable communication between computers on the network, then follow this procedure:

1. Unplug everything from the network, and make a very small network comprising one BBC Microcomputer, the Econet clock, a File Server and one terminator, with the necessary leads and adaptors to connect them all together. Open the clock box and turn on PERIOD switches 4 and 5 (marked 2 us and 4 us respectively) and turn on MARK switch 4 (marked 1 us). All other switches should be off.

2. Hold down letter N on the BBC Microcomputer, and press and release <Break>, then release N. This should appear at the top of the screen

```
BBC Microcomputer
```

```
Econet station nnn
```

```
BASIC
```

```
>
```

If the second line reads **Econet station nnn No clock**, then suspect the clock, but first try the same experiment with different connecting leads and a different BBC Microcomputer. Check the clock by plugging it directly into the Econet socket of the File Server and find whether the **No clock** light goes out.

3. If the BBC Microcomputer does not give a **No clock** message, then try logging on to the File Server with the command **\*I AM 0.254 <user id.>**. If there is a pause of about a minute, followed by **Not listening** or **Line jammed**, then continue on to step 4. If there is a prompt **>** after a short period, or an error message like **User not known** or **Incorrect password**, then you can conclude that the File Server and one BBC Microcomputer are functioning correctly, and proceed to step 7.

4. Reset the File Server station number to 254 as described in Section 7.4 (This step does not apply to RM380Z File Servers, but the Econet card in the RM380Z should be checked, to make sure that its station identifier links are correct.) Try logging on again as described in step 3 -- if this now functions, the File Server real-time clock had failed, either because the unit had not been switched on for more than about 6 months, or because of a hardware fault (contact your dealer or SJ Research in this case). Note that you will have to reset the time (and on the FDFS, the Baud rate) after this step. If the File Server now functions, proceed to step 7, otherwise continue.

5. Check the installation and operating instructions for the File Server to check that all has been done correctly. If you still cannot log on, then contact your dealer or SJ Research for advice.

6. It is still possible to check the network communication without a File Server. You will have to key in the following short program, which checks that the network and Econet interfaces function correctly. It can also be used to check an Acorn Level 2 File Server hardware, by stopping the program, then pressing <Ctrl-Break> on the File Server computer.

Key in this program (you do not of course need the REM statements):

```

10REM Program to test Econet
20
30 DIM X% 20
40REPEAT
50 INPUT "Station to test : "stn%
60 Y%=X% DIV 256
70 ?X%=1
80REM this is the reason code for 'send string'
90 X%?1=stn% : X%?2=0
100REM X%?2 = network number for multiple networks
110 $(X%+3)="*| Are you there ??"
120 A%=&14
130 CALL &FFF1 : REM Osword
140UNTIL FALSE

```

It would be wise to save this program, and also NETMON and STATIONS (if you have managed to get them off the File Server) on to local disc or cassette.

The program will repeatedly prompt for station numbers, and will execute the same call that \*NOTIFY uses, sending string **Are you there ??** to each station specified. The target station must have its protection byte *unset* -- the easiest way to do this on a BBC microcomputer is to press <Ctrl-Break> on every station. For Masters and Econet Terminals the \*UNPROT command is available.

Start with only two BBC Microcomputers on a short network with just the clock, one terminator, and connecting leads and adaptors. Check that you can send the message from one to the other. Sending the message only one way in fact causes a bi-directional communication, so fully tests both network interfaces.

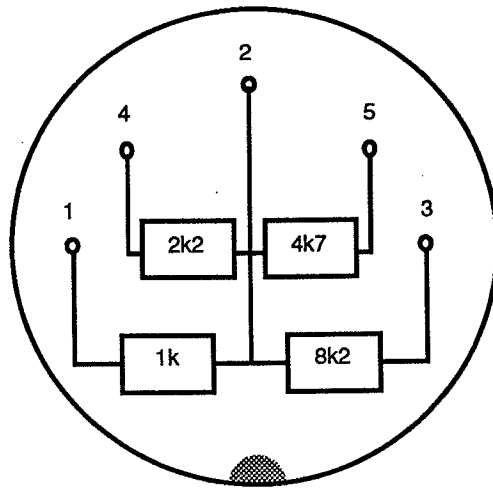
Possible errors are (note that some of these errors only appear after the computer has been trying for about a minute):

No clock	You should have checked this already by pressing <N-Break> on each computer (see step 2).
Net error or Line jammed	First try unplugging the terminator, and replacing it with the other one. If this does not cure the problem, then try replacing the connecting leads one at a time. Finally, replace the BBC Microcomputers one at a time. If you still cannot make contact between any two computers, check that you do not have any consistency errors: check that your connecting leads are wired pin 1 to pin 1 etc. (it is easiest to do this with a multimeter and an assistant), and ensure that whoever did the Econet upgrades to your BBC Microcomputers ran the full test (using a special Econet tester). If the network has worked in the past, it is possible that a lightning strike or large transient mains voltage pulse has destroyed all the SN75159 Econet line driver chips -- replace one or two and try the test again.
Not listening	Check first that the protection byte is not set, by typing <Ctrl-Break> on the destination computer, and re-run the program. If it still fails, follow the instructions under Net error.

If some or all of the BBC Microcomputers fail to communicate, then try new SN75159 Econet line driver chips (the chip should be in a socket). They can blow if there is a large transient voltage pulse on the network, such as that due to a local lightning strike.

To help avoid a repetition of this, SJ Research sell transient suppressor boxes to absorb transients of this type, and protect the computers.

7. If some or all of the BBC Microcomputers function in the very short network, but not in the main network, then first check that the network connections are all correct. The easiest way to do this is with a multimeter (set to an ohms range) and a special DIN plug wired with resistors as shown in Fig. 1 below.



**Fig 1 Test Plug (viewed from solder side)**

Unplug the clock, terminators and all computers (including the File Server). Plug the test plug into a socket in the network. At each socket outlet on the network poke one of the prods into the centre pin (pin 2) of the DIN socket. Then check that the resistances are as follows:

pin 1 to pin 2 1k  
pin 4 to pin 2 2k2  
pin 5 to pin 2 4k7  
pin 3 to pin 2 8k2

If any of these measured resistances differ significantly from these values, suspect an open or short circuit in the network. Remember that the network is split in the middle at the clock box, so the test plug will have to be plugged in on one side, and then on the other.



# Chapter 10 : Machine Code Reference Section BBC MOS Interface

This section is intended as a full and detailed handbook of the machine code interface to the network filing system. It covers the filing system interface, which is (in theory at least) the same as the interface to disc, tape or any other filing system. It also covers the network communication functions which are network specific.

The section assumes knowledge of the BBC operating system, BASIC (including the indirection operators - see chapter 39 of the BBC User Guide), assembler and control blocks.

Because this section is a reference section, it is not necessary to understand all of it at once. Instead it should be possible to look up a certain call as it is needed.

## Network Versions

There are three main Network filing system ROMs available for the BBC computer. These are :-

NFS version 3.34

NFS version 3.60

Advanced NFS (Only available for the Master, Econet Terminal & Compact)

(see OSARGS A=2 Y=0 and \*HELP for testing the version)

Although they are intended to control the same operations there are some important differences. Because all versions are common, programs should be written so that they will work on all.

## Memory Addresses

The filing system assumes that addresses in a remote or local machine are 4 bytes long. For a normal BBC computer this means that the top 16 bits are not relevant, but when a 2nd processor is used only the address &FFFFxxxx specifies the I/O processor. For the 6502 2nd processor the address &0000xxxx specifies the language processor. In general if the I/O processor is the intended destination then &FFFFxxxx should always be used. Unless otherwise specified, multi-byte numbers are stored low byte first.

For an Acorn Master & Acorn Econet Terminal the I/O processor is located at address &FFFFxxxx and the screen RAM is located at address &FFFExxxx.

## Calling Operating System Routines

These routines handle all the input and output to the Econet system. All routines require the Decimal flag (D) to be clear on entry. The Interrupt flag (I) is always preserved (but interrupts may be enabled during the call).

Some operating system routines require a control block; a small area of memory set up to hold the parameters to be used by the routine. An area of memory must be preserved for the control block, filled with the parameters you want, and pointers to the area given to the routine.

The information for most calls is described here in the form of a control block. The number on the left hand side is referring to the offset from the start of the control block. To set up a control block an area of memory should be reserved before the call is performed. It is simple to use the BASIC indirection operators to set up the values in the control block.

The program listed below performs a peek (i.e. a transfer of data from the memory of a remote machine to the memory of the local machine) by using the following control block :-

0	&81
1	0
2	Remote station (station number, network)
4	Pointer to start of local buffer
8	Pointer to end of local buffer
12	Pointer to start of remote machine's buffer
16	

```

10 DIM blk% 15:      REM Reserve a block of data
20 INPUT "Station : "station%
30 blk%?0=&81:      REM Control byte for peek
40 blk%?1=0:      REM Port number for immediate operation
50 blk%!2=station%: REM Insert station number
60 blk%!4=&FFFF7C00: REM Start of MODE 7 screen (not scrolled)
70 blk%!8=&FFFF8000: REM End of MODE 7
80 blk%!12=&FFFF7C00: REM Start of MODE 7 in remote machine
90
100 X%=blk%:      REM X points to low byte
110 Y%=X% DIV 256: REM Y points to high byte
120 A%=&10:      REM Set accumulator
130 CALL &FFF1:  REM Call OSWORD

```

Note that the above program does not check that the transmit operation worked. For a complete description of the peek operation see sections 10.9 and 10.14.

## Summary of all calls for Econet

Name	Call Address	Section	Description
OSGBPB	&FFD1	10.1	Read/write group of bytes to a file, read user's environment
OSFIND	&FFCE	10.2	Open/close file for random access
OSBGET	&FFD7	10.3	Get byte from file (Use OSGBPB if possible)
OSBPUT	&FFD4	10.4	Write byte to file (Use OSGBPB if possible)
OSARGS	&FFDA	10.5	Read/write file's arguments (using a handle)
OSFILE	&FFDD	10.6	Read/write file's arguments
OSCLI	&FFF7	10.7	Send string to command line interpreter
OSWORD	&FFF1	10.8	Network Specific commands
		10.9	Transmit data (A=&10)
		10.10	Receive data (A=&11)
		10.11	Read arguments (A=&12)
		10.12	Read/write station information (A=&13)
		10.13	Send to File Server (A=&14)
OSBYTE	&FFF4	10.14	Poll transmission and reception
		10.15	Transmitting
		10.16	Receiving
		10.17	Port numbers
		10.18	The Bridge
		10.19	Printers
		10.20	The File Server Interface
		10.21	Password entry format
		10.22	Application Notes
		10.23	Netmon
		10.24	Econet Protocols

## 10.1 OSGBPB

## Call Summary

Entry point &FFD1  
Indirected via &21A

### On entry,

A=Reason code  
YX points to a control block 13 bytes long.

Do not put the control block in page zero. There is a bug in NFS 3.34 such that OSGBPB will not work at all if this is done.

Value in A	Function
A=1	Write block using offset
A=2	Write block ignoring offset
A=3	Read block using offset
A=4	Read block ignoring offset
A=5	Read currently selected disc title and boot option
A=6	Read directory
A=7	Read library
A=8	Read specified number of file names

### On exit,

A = Return code indicating whether the requested function is supported by the currently selected filing system. If A=0 then the operation was attempted. A is returned unchanged if that function is not available.  
YX undefined

Control block is modified on completion. On some filing systems the carry flag is set on reaching the end of the file/directory, this should not be relied on. To write software that is compatible with all filing systems the only method of finding whether all the files have been read is to look at the contents of (control block + 4).

## Write data giving offset

### General description

This call writes a block of data from RAM to a file, specifying where in the file to put the bytes.

### On entry,

A=1

YX point to the control block shown below :-

0	File Handle
1	Address of data
5	Number of bytes to transfer
9	Offset in file
13	

### On exit,

The sequential pointer (PTR#) will have been updated to point immediately beyond the last byte written. If the operation succeeded the control block is :-

0	File Handle
1	Old address + Number of bytes transferred
5	Number of bytes not transferred (should=0)
9	Old offset + Number of bytes transferred
13	

### Example :

The procedure below uses this call to write a string to the file opened as handle%.

```
DEF PROCwrite_string(handle%,$data%,offset%)
LOCAL A%,X%,Y%
blk%?0=handle%
blk%!1=data%
blk%!5=LEN($data%)
blk%!9=offset%
X%=blk%
Y%=X% DIV 256:A%=1
IF (USR(osgbpb) AND &FF)<>0 THENPROCsimulate_osgbpb(blk%)
ENDPROC
```

### OSGBPB A=1

The procedure below can be used on systems that do not support OSGBPB (e.g.TAPE).

```
DEF PROCsimulate_osgbpb(blk%)
LOCAL I%
PTR#?blk%=blk%!9
FOR I%=blk%!1 TO blk%!1+blk%!5
BPUT#?blk%,?I%
NEXT
ENDPROC
```

The following program uses the above procedure to create a file called 'filename' and write two strings to it.

```
10 DIM data% 255 :REM Maximum size of my string (actually 256 bytes)
20 DIM blk% 12 :REM My control block
30 osgbpb=&FFD1
50 out_ch%=OPENOUT"Filename"
60 PROCwrite_string(out_ch%,"This is a test",0)
70 PROCwrite_string(out_ch%,"Another piece of text",&22A)
80 CLOSE#out_ch%
90 END
```

### Compatibility between Filing Systems :

Not supported on TAPE, ROM or some non-Acom DFS. Note on the Acom File Server a write beyond the end of the file will give an EOF (end of file) error.

## Write Data

OSGBPB A=2

### General Description

This call writes a block of data to a file using the current pointer (i.e. the value returned by PTR#).

### On entry,

A=2

YX point to the control block shown below :-

0	File Handle
1	Address of data
5	Number of bytes to transfer
9	Ignored
13	

### On exit,

The sequential pointer (PTR#) will have been updated to point immediately beyond the last byte written. If the operation succeeded the control block is :-

0	File Handle
1	Old location + Number of bytes transferred
5	Amount of bytes not transferred (should=0)
9	Corrupt
13	

### Compatibility between Filing Systems :

Not supported on TAPE or ROM or some non-Acom DFS.

## Read Data giving offset

OSGBPB A=3

### General description

This call reads a block of data from a file into RAM, from a specified offset in the file (i.e. ignoring PTR#). On reaching the end of the file the carry flag (C) is set, and the amount of data not transferred is returned in the control block. An EOF (End of file) error will be returned if the offset is past the end of the file. If the offset of a file is equal to the extent, then no bytes will be transferred.

### On entry,

A=3

YX point to the control block shown below :-

0	File Handle
1	Location to put the data
5	Number of bytes to transfer
9	Offset in file
13	

### On exit,

If the operation was successful the control block is :-

0	File Handle
1	Old location + Number of bytes transferred
5	Number of bytes not transferred
9	Old offset + Number of bytes transferred
13	

The number of bytes not transferred will be zero unless the end of file has been reached; in which case all bytes from the offset up to the end of the file will have been transferred, and the number not transferred is the difference between the number originally asked for and the number actually transferred. Note that an area of memory large enough to hold the entire quantity of data asked for will always be corrupted, so it is inefficient to request many more bytes than are expected to be available.

NFS 3.6 does NOT return the correct result for EOF# after this call. The only way of checking to see if the End Of File has been reached, is to read the result from the control block (bytes 5-8 inc.).

### Compatability between Filing Systems :

Not supported on TAPE or ROM or some non-Acom DFS. Note that the Acom DFS does not corrupt more memory than is required to hold the data loaded, whereas the NFS will always transfer the amount of data requested. Therefore programs which request more bytes than there is memory space to hold may work on the DFS but crash with NFS.

## Read Data

### General description

This call reads a block of data to a file, at the position given by the sequential pointer (PTR#).

### On entry,

A=4

YX point to the control block shown below :-

0	File Handle
1	Address of data
5	Number of bytes to transfer
9	No data needed but will be modified
13	

### On exit,

If the operation was successful the control block is :-

0	File Handle
1	Old location + Number of bytes transferred
5	Number of bytes not transferred
9	Corrupt
13	

N.B.

Number of bytes not transferred and memory corruption as for OSGBP B A=3.

### Compatibility between Filing Systems :

Not supported on TAPE or ROM or some non-Acorn DFS. Note that not all filing systems return C=1 to indicate that the end of file has been reached. It is strongly recommended that to find whether the end of file has been read the number of bytes not transferred is checked for a non-zero value.

OSGBP B A=4

## Read CS disc name

### General description

This call returns your currently selected disc name and your boot option.

### On entry,

A=5

YX point to the control block shown below :-

0	xxx
1	Address where data will be loaded
5	xxx
13	

xxx indicates that no parameter is required.

### On exit,

C undefined

The control block is left unchanged.

The address pointed to by the control block is modified as shown below :-

0	Length of disc title (n)
1	Disc title (Max.16 characters)
(n+1)	Your boot option as set by *OPT4
(n+2)	

The disc name is normally 16 characters padded with spaces. It is possible for the disc name to have spaces as significant characters, e.g. 'Wombat 2' (although on the network it would not be possible to use this name). Therefore to read the disc title the name should be read backwards, stripping off spaces until the first non-space character.

### Compatibility between Filing Systems :

Not supported on TAPE or ROM.

OSGBP B A=5

## Read CSD name

### General description

This call returns your currently selected directory name.

### On entry,

A=6, YX point to the control block shown below :-

0	Undefined
1	Address to put the data
5	Undefined
9	Undefined
13	Undefined

There is a bug in NFS version 3.34 which sometimes causes the error 'No reply'. To get around this problem the following fix should be used before calling OSGBPB.

```
.fix_bug  LDA #2:LDY #0:JSR &FFDA \Call OSARGS to read version of NFS
          CMP #2:BNE not_required \Fix not required if not version 3.34
          LDX #block MOD_256      \Fix NFS workspace by writing
          LDY #block DIV 256      \ an &CB to location &B8
          LDA #6:JSR &FFF1        \OSWORD to write to I/O processor
.not_required
```

(Rest of the code)

```
.block    EQU &FFFF00B8          \Write &CB to location &FFFF00B8
          EQU &CB
```

N.B. It is essential that the version is checked before modifying this location, because it cannot be guaranteed that the use of this location will not change.

## OSGBPB A=6

### On exit,

The address pointed to by the control block is modified as shown below :-

0	Length of drive number(v)=0 on Econet
1	ASCII coded drive number This field is not present on the Econet
v	Length of directory name (n) (Usually 10 characters)
(v+1)	Directory name
(n+2)	Ownership (0=Owner,&FF=public)
(n+3)	

The directory title may be padded with spaces.

### Example :

The following program will print out the currently selected directory, and drive number, on all filing systems that support the call.

```
10 REM   Read currently selected directory name
20 REM   This program is compatible with all filing systems
30 REM   (C) A.J.Engeham, SJ Research
40 REM
50
60 osgbpb=&FFD1
70 DIM blk% 17,buffer% 100
80 blk%?0=0
90 blk%!1=buffer%
100 blk%!5=0
110 blk%!9=0
120 X%=blk%:Y%=X% DIV 256
130 A%=6
140 IF (USR(osgbpb) AND &FF)=6 THENP."Not supported":END
150 pos%=0
160 PROCread("Current drive number is : ")
170 PROCread("Currently selected directory is : ")
180END
190
200DEF PROCread(string$)
210 LOCAL V%,result$,I%
220 V%=buffer%?pos%
230 pos%=pos%+1
240 IF V%=0 THENENDPROC
250 FOR I%=pos% TO pos%+V%-1
260 result$=result$+CHR$(I%?buffer%)
270 NEXT
280 pos%=pos%+V%
290 PRINTstring$;result$
300ENDPROC
```

### Compatibility between Filing Systems :

Not supported on TAPE or ROM. Note that early versions of ADFS do not return the ownership byte.

## Read LIB name

OSGBPB A=7

### General description

This call returns your currently selected Library.

### On entry,

A=7

YX point to the control block shown below :-

0	Undefined
1	Address to put the data
5	Undefined
9	Undefined
13	

### On exit,

The address pointed to by the control block is modified as shown below :-

0	Length of drive number(v)=0 on Econet
1	ASCII coded drive number This field is not present on the Econet
v	Length of Library name (n)
(v+1)	Library name
(n+1)	Ownership (0=Owner,&FF=public)
(n+2)	

### Example :

See OSGBPB with A=6

The Library name may be padded with spaces.

### Compatibility between Filing Systems :

Not supported on TAPE or ROM.

# Read objects

## General description

This call reads file/directory names from your currently selected directory.

## On entry,

A=8

YX point to the control block shown below :-

0	0
1	Location to put the data
5	Maximum number of file names to transfer
9	Which file to start on
13	

## On exit,

The control block is modified as shown below.

0	Cycle number
1	Pointer to the end of data
5	Number of files NOT transferred
9	Which file to ask for next
13	

The file names are padded with spaces.

0	Length of filename (n)
1	Filename (padded with spaces)
(n+1)	Length of filename(if asked for)
(n+2)	Filename
	.
	.
	.
v	

## OSGBPB A=8

## Example

The program shown below will print out all the file names of your currently selected directory. Note that, because the starting file is the first one in the directory, this program will produce the same results for the Disc Filing System as the Network Filing System.

```
DIM blk% 13,buffer% 20
blk%!9=0:REM Start at the first file
REPEAT
blk%?0=0:REM Cycle number returned here
blk%!1=buffer%
blk%!5=1:REM Number of files to read
X%=blk%:Y%=blk% DIV 256
A%=8:CALL &FFD1
IF blk%!5=0 THEN?(buffer%+1+buffer%?0)=13:PRINT $(buffer%+1)
UNTIL blk%!5=1
```

## Compatibility between Filing Systems :

Not supported on TAPE or ROM. Entry number are not guaranteed to be sequential; the only value which should be used in general are 0 and values returned by a previous call. In the special case of Econet, entry numbers start from zero and ascend in steps on 1 to specify objects in alphabetical order.

The carry flag is NOT valid on exit. The DFS is misleading in this respect, although C happens to be significant on DFS systems, it is not an official Acorn feature.

## 10.2 OSFIND

Entry point &FFCE  
Indirected via &21C

### On entry,

A=Type of operation.

### On exit,

A=file handle if successful.  
A=0 not possible to open file.  
C,N,V,Z are undefined.  
Contents of control block are preserved.

#### Value in A

A=0

A=&40

A=&80

A=&C0

#### Function

Close file specified by Y register

Open file for input

Open file for output

(i.e. create new file deleting old file)

Open file for update (i.e. input and output).

## Call Summary

## Close files

### General description

If Y=0 then the call closes all open files but will not close directory context handles.  
If Y<>0 then the call closes the file handle, or context handle, held in Y.

This call is the equivalent to CLOSE#Y in BASIC.

### On entry,

A=0

Y=handle to close (0 to close all files)

### On exit,

A,X,Y undefined

### Compatibility between Filing Systems :

Supported on all filing systems. Y=0 does not work on some version of Master DFS.

OSFIND A=0

## Open file for input

OSFIND A=&40

### General description

Returns a handle for a file whose name is in memory, pointed to by YX. This handle can then be used for reading bytes from that file. The handle can also be used as a context handle to describe your environment on your File Server. This call is equivalent to the BASIC 2 keyword OPENIN.

### On entry,

YX point to a control block containing a string terminated by a carriage return (&0D).  
A=&40

### On exit,

A=file handle if successful  
A=0 if unable to open file  
X,Y undefined  
C,N,V,Z undefined

### Compatibility between Filing Systems :

Supported on all filing systems.

## Open file for output

OSFIND A=&80

### General description

Create a file with the name given at memory address YX and returns a handle for that file. The file is opened for both reading and writing. Any previous file of that name is deleted. This is equivalent to the BASIC keyword OPENOUT.

### On entry,

YX point to a filename terminated by a carriage return (&0D).  
A=&80

### On exit,

A=file handle if successful  
A=0 if unable to open file  
X,Y undefined  
C,N,V,Z undefined

### Compatibility between Filing Systems :

Not supported on ROM, or any other read only filing systems.

## Open file for update

OSFIND A=&C0

### General description

Returns a handle for a file whose name is pointed to by YX. The file is opened for both reading and writing. This is equivalent to the BASIC 2 keyword OPENUP or the BASIC 1 keyword OPENIN.

### On entry,

YX point to a control block terminated by a carriage return (&0D).  
A=&C0

### On exit,

A=file handle if successful  
A=0 if unable to open file  
X,Y undefined  
C,N,V,Z undefined

### Example

The example shows how to OPENUP a file for BASIC 1 and BASIC 2.

```
DIM dummy% 80
$dummy%=""
INPUT "Filename : "dummy$
P.FNopenup(dummy$)
END
```

```
DEF FNopenup($dummy%)
  LOCAL A%,X%,Y%
  X%=dummy%
  Y%=dummy% DIV 256
  A%=&C0
  =(USR(&FFCE) AND &FF)
```

### Compatibility between Filing Systems :

Not supported on ROM. TAPE treats the open for update as an open for output (i.e. OSFIND with A=&80).

## 10.3 OSBGET

## Read Byte

Entry point &FFD7  
Indirected via &216

### General description

This call reads one byte from a file, using a handle returned from an OSFIND call. The file pointer (PTR#) is incremented.

### On entry,

Y=file handle

### On exit,

X,Y preserved  
If successful A=byte retrieved from file, C=0  
If the end of the file is reached A=254, C=1  
If an attempt to read past the end of file is made, an 'EOF' error will be returned.

### Compatibility between Filing Systems :

Supported on all filing systems. Although this call is supported on the Econet it is very slow, it is essential that wherever possible OSGBPB is used. This is because the file server takes time to process each byte, and time is taken to send each byte across the network; if the file server (if it isn't one of our multitasking file servers) is processing someone else's command your machine will wait in the mean time. If \*PUTGET is active, or your machine has ANFS, then the indirection vector will be modified so that the data is read from the fileserver using OSGBPB, and later retrieved from a buffer in memory.

## 10.4 OSBPUT

Entry point &FFD4  
Indirected via &218

### General description

This call puts one byte to a file, using a handle returned from an OSFIND call. The file pointer (PTR#) is incremented.

### On entry,

Y=file handle  
A=byte to put

### On exit,

A,X,Y are preserved  
N,V,Z,C are undefined

### Compatibility between Filing Systems :

Supported on all filing systems except ROM, and other read only filing systems. Although this call is supported on the Econet it is very slow, it is essential that wherever possible OSGBPB is used. This is because the file server takes time to process each byte, and time is taken to send each byte across the network; if the file server (if it isn't one of our multitasking file servers) is processing someone else's command your machine will wait in the mean time. If \*PUTGET is active, or your machine has ANFS, then the indirection vector will be modified so that the data is read from the fileserver using OSGBPB, and later retrieved from a buffer in memory.

## Write Byte

## 10.5 OSARGS

Entry point &FFDA  
Indirects via &214

### On entry,

Value in Y	Value in A
------------	------------

Y=0	A=0
Y=0	A=1
Y=0	A=2

Y<0	A=0
Y<0	A=1
Y<0	A=2
Y<0	A=3
Y<0	A=4

Y<0	A=&80
	A=&FF

### On exit,

A=0 if the operation supported (except A=0, Y=0) otherwise A is preserved  
X,Y are preserved  
C,N,V,Z are undefined

## Call Summary

Function
----------

Return filing system type
Return rest of command line
Return version of NFS ROM

Read sequential pointer
Write sequential pointer
Read extent of file
Write extent of file (ANFS only)
Read current amount of space allocated to file (ANFS only)
Read internal information (ANFS only)
Ensure file is up to date on the media

## Return Filing System Type

### General description

This call returns, in A, the type of your current filing system.

### On entry,

A=0  
Y=0

### On exit,

A=filing system  
X,Y undefined  
N,V,Z,C undefined

Value in A	Meaning
0	No current filing system
1	Cassette, 1200 baud
2	Cassette, 300 baud
3	ROM
4	Disc
5	Econet
6	Teletext/Prestel
7	IEEE
8	Advanced Disc
9	Host
10	VFS Video disc
16	Acacia RAM

To select filing system 'n' use OSBYTE with A=&8F, X=&12, Y=n.

### Compatibility between Filing Systems :

Supported on all filing systems.

OSARGS A=0 Y=0

## End of line parameter

### General description

This call set the zero page locations (pointed to by X on entry to the call) to point to the parameter at the end of the last command line.

Unfortunately, there is a bug in version 3.34 of the NFS ROM such that the call returns the pointer pointing to the text immediately after the last \*. For example, in the command line '\*\*\*\*\*VIEW 23' it would point to the 'V' in VIEW. On all other filing systems that support this call, and later versions of the NFS ROM, the call returns the pointer pointing to the '2' in the 23.

### On entry,

A=1  
Y=0  
X=Zero page location

### On exit,

A=0 if call supported otherwise A is preserved.  
X,Y preserved  
N,V,Z,C undefined

### Example :

The program below shows a typical method of getting round the problem with different versions of the NFS.

```
osargs=&FFDA

.error    BRK                \ 'entry' is the start address
          EQUB 0              \ Your error number
          EQU$ "Syntax : <correct syntax>":BRK \ Your error message
.entry    LDA #1:LDY #0:LDX #zp \ zp is a zero page location
          JSR osargs           \ Get command line
          LDY #0:LDA #0:JSR osargs \ Find what filing system
          CMP #5:BNE new_nfs    \ It's o.k. if the FS isn't Econet
          LDA #2:LDY #0:JSR osargs \ Read version of NFS
                                   \ (See OSARGS A=2 Y=0)
          CMP #2:BNE new_nfs    \ Branch if the NFS is >3.34
.loop1    INY:LDA (zp),Y
          CMP #&D:BEQ error      \ If no parameter supplied then error
          CMP #ASC" ":BNE loop1 \ Repeat until space is found
.loop2    INY
          LDA (zp),Y
          CMP #ASC" ":BEQ loop2 \ Deal with spaces between
                                   \ command and parameter
.new_nfs   Your program starts here!
          \ Next letter is at (zp),Y
```

The above program will only work if executed in the I/O processor. If this is to be tube compatible then the block of memory, holding the end of the command line parameter, must be transferred across the tube.

### Compatibility between Filing Systems :

Not supported on TAPE or ROM.

## Version of NFS

OSARGS A=2 Y=0

### General description

This call returns the version of NFS ROM.

### On entry,

A=2  
Y=0

### On exit,

A=Version of NFS  
X,Y preserved  
N,V,Z,C undefined

Value in A	Meaning
A=1	ANFS, NFS version 3.40 or greater
A=2	NFS version 3.34 or call not supported

### Compatibility between Filing Systems :

Supported on NFS only. If you are trying to write programs which are to be compatible on all filing systems then you must check to see if the current filing system is the network (OSARGS with A=0, Y=0) before issuing this call. For example, if the call is being used to get around the bugs in NFS version 3.34 then first check the filing system, if the filing system is not network then do not use this call, because the result of the call will be A=2 (call not supported).

## Read PTR#

OSARGS A=0 Y<>0

### General description

Read the sequential pointer (PTR#) leaving the result in four consecutive zero page locations specified in X.

#### On entry,

A=0  
Y=file handle (see OSFIND)  
X=zero page location

#### On exit,

A=0  
X,Y preserved  
N,V,Z,C undefined

The address pointed to by X (zero page location) holds the sequential pointer in 6502 order as a 4 byte value.

### Compatibility between Filing Systems :

Not supported on TAPE or ROM.

## Write PTR#

OSARGS A=1 Y<>0

### General description

Write the sequential pointer (PTR#) of a file using the value held in four consecutive zero page locations pointed to by X.

#### On entry,

A=1  
Y=file handle (see OSFIND)  
X=zero page location

#### On exit,

A=0 if operation successful, otherwise A=1  
X,Y preserved  
N,V,Z,C undefined

The contents of the zero page locations are preserved.

### Compatibility between Filing Systems :

Not supported on TAPE or ROM.

## Read extent

### General description

Reads the extent of a file leaving the result in four consecutive zero page locations specified in X.

### On entry,

A=2  
Y=file handle (see OSFIND)  
X=zero page location

### On exit,

A=0 if operation successful, otherwise A=2  
X,Y preserved  
N,V,Z,C undefined

The address pointed to by X (zero page location) holds the extent of the file in 6502 order as a 4 byte value.

### Compatibility between Filing Systems :

Not supported on TAPE or ROM.

OSARGS A=2 Y<>0

## Write extent

### General description

Writes the extent of a file (equivalent to BASIC's EXT#) using the value given in four consecutive zero page locations specified in X.

### On entry,

A=3  
Y=file handle (see OSFIND)  
X=zero page location

### On exit,

A=0 if operation successful, otherwise A=3  
X,Y preserved  
N,V,Z,C undefined

The contents of the zero page locations are preserved.

### Compatibility between Filing Systems :

This call is only currently supported on the Advanced Network filing system ROM. However a call of this nature requires cooperation from the file server as well, and not all versions of File Server support it either.

OSARGS A=3 Y<>0

## Read space

### General description

Reads the amount of space currently allocated to a file.

### On entry,

A=4  
Y=file handle (see OSFIND)  
X=zero page location

### On exit,

A=0 if operation successful, otherwise A=4  
X,Y preserved  
N,V,Z,C undefined

### Compatibility between Filing Systems :

This call is only supported on the Advanced Network Filing system ROM.

OSARGS A=4 Y<=>0

## ANFS Internal Info.

### General description

Return internal information held by the Advanced Network filing system ROM.

### On entry,

A=&80  
Y=handle to return information on.  
X points to four bytes in zero page.

### On exit,

The first byte has the handle in file server format.  
The second and third byte hold the fileservr number.  
The fourth byte holds the following information :-

Bit 0 is the sequence number  
Bit 1 is the 'known to be a dir' flag bit  
Bit 2 is the 'thought to be URD' bit  
Bit 3 is the 'thought to be CSD' bit  
Bit 4 is the 'thought to be CSL' bit  
Bit 5 is the 'current context' bit  
Bit 6 is the EOF error bit, set if the next BGET will fail  
Bit 7 is the write flag, set means writable to

### Compatibility between Filing Systems :

This call is only supported on the Advanced Network Filing system ROM.

OSARGS A=&80 Y<=>0

## Update file

### General description

Ensures that any data held temporarily in RAM is transferred to disc. Useful in programs which may run for many hours and do not wish to risk losing data if the power fails etc.

### On entry,

A=&FF

### On exit,

A=0 if operation successful, otherwise A=&FF

X,Y preserved

N,V,Z,C undefined

### Compatibility between Filing Systems :

Not supported on TAPE or ROM. Although this call is supported on the standard version of the NFS ROM the call has no effect because the data will always be up to date on the file server; this is not the case with the ANFS ROM because OSBGET and OSBPUT are buffered internally.

OSARGS A=&FF

## 10.6 OSFILE

Entry point &FFDD

Indirected via &212

### On entry,

A=Reason code

YX point to the control block

Value in A

Function

A=0

Save a block of memory as a file using the information provided in the parameter block.

A=1

Write the information in the parameter block to the catalogue entry for an existing file (i.e. file name and address).

A=2

Write the load address (only) for an existing file.

A=3

Write the execution address (only) for an existing file.

A=4

Write the attributes (only) for an existing file.

A=5

Read a file's catalogue information, with the file type returned in the accumulator. The information is written to the parameter block.

A=6

Delete the named file.

A=7

Create a file (ANFS only)

A=&FF

Load file to given address.

## Call Summary

## Save a file

### General description

This call saves a block of memory as a file, using the information provided in the control block.

### On entry,

A=0

YX point to the control block shown below :-

0	Address of filename
2	Reload address
6	Execution address
10	Start address of data
14	End address of data
18	

### On exit,

A,X,Y,C undefined

The control block becomes corrupt.

### Compatibility between Filing Systems :

Not supported on read only systems such as ROM filing system.

OSFILE A=0

## Change file info.

### General description

This call is used to change the reload address, execution address and attributes of a file to the values given.

### On entry,

A=1

YX point to the control block shown below :-

0	Address of filename
2	Reload address
6	Execution address
10	Attributes
11	

The attributes of a file are 8 bits corresponding to the following values :-

Bit	Access set on Econet	Meaning if set
7		Undefined on network (locked for other users on some filing systems)
6		Undefined on network (executable by other users on some filing systems)
5	W public access	The file is writeable by other users
4	R public access	The file is readable by other users
3	L	The file is locked
2	M	Undefined on network (executable by you on some filing systems)
1	W owner access	The file is writeable by you
0	R owner access	The file is readable by you

### On exit,

A,X,Y,C undefined

### Compatibility between Filing Systems :

Not supported on TAPE or read only filing systems such as ROM.

OSFILE A=1

## Change reload address

OSFILE A=2

### General description

This call is used to change the reload address of a file to that specified in the control block.

### On entry,

A=2

YX point to the control block shown below :-

0	Address of filename
2	
6	Reload address

### On exit,

A,X,Y,C are undefined

### Compatibility between Filing Systems :

Not supported on TAPE or read only filing systems such as ROM.

## Change execute address

OSFILE A=3

### General description

The call is used to change the execution address of a file to that given in the control block.

### On entry,

A=3

YX point to the control block shown below :-

0	Address of filename
2	
6	Undefined
10	Execution address

### On exit,

A,X,Y,C undefined

### Compatibility between Filing Systems :

Not supported on TAPE or read only filing systems such as ROM.

## Change attributes

### General description,

This call is used to change the attributes of a file or directory.

### On entry,

A=4

YX point to the control block shown below :-

0	Address of filename
2	
14	Undefined
15	File attributes

See OSFILE A=1 for meaning of file attribute bits.

### On exit,

A,X,Y,C undefined

### Compatibility between Filing Systems :

Not supported on TAPE or read only filing systems such as ROM.

## OSFILE A=4

## Read catalogue info.

### General description

This call is used to read a file or directory's catalogue information into a control block.

### On entry,

A=5

YX point to the control block shown below :-

0	Address of filename
2	
15	undefined

### On exit,

X,Y,C undefined

A=0 if object not found, A=1 if object was a file, A=2 if object was a directory.

0	Address of filename
2	
6	Load address
10	Execution address
14	Length
15	Attributes
17	Creation date
18	Undefined

The attributes are 8 bits corresponding to the following values :-

Bit	Access set on Econet	Meaning if set
7	L	Locked for other users
6	R public access	Executable by other users
5	W public access	Writeable by other users
4	R public access	Readable by other users
3	L	Locked for you
2	R owner access	Executable by you
1	W public access	Writeable by you
0	R owner access	Readable by you

### Compatibility between Filing Systems :

Not supported on TAPE or ROM.

## OSFILE A=5

## Delete object

### General description

Deletes the filename pointed to by the control block.

### On entry,

A=6

YX point to the control block shown below :-

0	Address of filename
2	
18	Undefined

### On exit,

X,Y,C undefined

If object not found then A=0 otherwise A>0 and control block updated as OSFILE A=5.

### Compatibility between Filing Systems :

Not supported on TAPE or read only filing systems such as ROM.

OSFILE A=6

## Create file

### General description

This call is the same as the save operation (OSFILE A=0) except that no data is transferred. Consequently the newly created file is filled with zeros.

### On entry,

A=0

YX point to the control block shown below :-

0	Address of filename
2	
6	Reload address
10	Execution address
14	Start address
18	End address

### On exit,

A,X,Y,C undefined

The control block becomes corrupt.

### Compatibility between Filing Systems :

Not supported on read only systems such as ROM filing system, nor on NFS 3.34/3.6, early versions of File Server, early versions of DFS.

OSFILE A=7

## Load file

### General description

This call loads a file at a specified reload address, or, if the low byte of the execution address parameter is not zero, at the file's own reload address. The control block is updated with the file's catalogue information (load address, execution address, length, attributes).

### On entry,

A=&FF

YX point to the control block shown below :-

0	Address of filename
2	
15	Undefined

### On exit,

A,X,Y,C undefined

0	Address of filename
2	
6	Reload address
10	Execution address
14	Length
15	Attributes

### Compatibility between Filing Systems :

Not supported on TAPE or ROM filing systems.

OSFILE A=&FF

## 10.7 OSCLI

Entry point &FFF7  
Indirected via &208

### General description

This call is used to send a command line to the Operating system or your currently selected filing system. This is the equivalent of a direct command with a '\*' in front of it.

### On entry,

A is undefined

YX point to the command line to be interpreted.

### On exit,

A,X,Y,C undefined

### Example :

```
10 DIM s% 80
20 PROCoscli("CAT")
30 END
40
50DEF PROCoscli($s%)
60 X%=s%:Y%=s% DIV 256
70 CALL &FFF7
80ENDPROC
```

If the procedure is used instead of the keyword OSCLI, then it can be guaranteed that this part of the program will not be sensitive to version 1 of BASIC.

## Command line

## 10.8 OSWORD

Entry point &FFF1  
Indirects via &20C

### On entry,

Value in A	Control byte	Function
A=&10	&81	READ a block of memory from a remote machine
A=&10	&82	WRITE a block of memory to a remote machine
A=&10	&83	CALL location and send argument block to remote machine
A=&10	&84	User procedure call to a remote machine
A=&10	&85	Operating system procedure call to a remote machine
A=&10	&86	HALT a remote machine
A=&10	&87	START a remote machine (after a Halt)
A=&10	&88	Read the machine type and version number of a remote machine
A=&11		Receive block operations
A=&12		Reading arguments
A=&13		Reading/writing station information
A=&14	&00	Communicate with File Server
A=&14	&01	Send Text message
A=&14	&02	Cause remote error

Calls with A=&10 are all variants of *Transmit* see section 10.15 for details of the necessary polling and retry techniques.

## Call Summary

## 10.9 Peek

### General description

This returns a block of memory from a remote machine into the local machine. See section 10.15 for information on polling the transmit for success.

### On entry,

A=&10

YX point to the address of the control block shown below :-

0	&81
1	0
2	Remote station (station number, network)
4	Pointer to start of local buffer
8	Pointer to end of local buffer
12	Pointer to start of remote machine's buffer
16	

### On exit,

A,X,Y undefined

0	Modified
1	Unchanged
16	

### Compatibility between Filing Systems :

Only supported by the NFS.

OSWORD A=&10 (Control byte=&81)

## Poke

OSWORD A=&10 (Control byte=&82)

### General description

This sends a block of memory from the local machine into the remote machine. See section 10.15 for information on polling the transmit for success.

### On entry,

A=&10

YX point to the address of the control block shown below :-

0	&82
1	0
2	Remote station (station number, network)
4	Pointer to start of local buffer
8	Pointer to end of local buffer
12	Pointer to start of remote machine's buffer
16	

### On exit,

A,X,Y undefined

0	Modified
1	Unchanged
16	

### Compatibility between Filing Systems :

Only supported by the NFS. The NFS need not be the current filing system.

## Remote JSR

OSWORD A=&10 (Control byte=&83)

### General description

This call sends an argument block to a remote machine, then jumps to a location in the remote machine. See section 10.15 for information on polling the transmit for success.

### On entry,

A=&10

YX point to the address of the control block shown below :-

0	&83
1	0
2	Remote station (station number, network)
4	Pointer to start of local buffer for arguments
8	Pointer to end of local buffer for arguments
12	Address to CALL in remote machine
16	

After this call the remote machine is protected against procedure calls and O.S. procedure calls until the parameter block is read. The program in the remote machine must read the parameter block (OSWORD A=&12) before exiting (with a RTS), otherwise the remote machine will remain protected. If the remote machine is a BBC microcomputer then the address to call must be in the I/O processor.

Although interrupts are disabled in the remote machine, they should be enabled if the routine is going to take much longer than 1 mS to complete.

The maximum size of the argument block is 128 bytes.

### On exit,

A,X,Y undefined

0	Modified
1	Unchanged
16	

### Compatibility between Filing Systems :

Only supported by the NFS. The NFS need not be the current filing system.

## Remote procedure

### General description

This passes a block of memory to a remote machine and causes an event (number 8) in that machine. The program in the remote machine must intercept the event number (procedure number held in YX) and read the argument block (OSWORD A=&12) before exiting with an RTS.

Note that the argument block must be read even if there are no arguments, because the Rx control block will not be reopened until this has happened.

Machines with Econet version 3.34 may crash, because stations greater than 240 can override the machine protection, therefore they can overwrite the argument block before it is read. See section 10.15 for information on polling the transmit for success.

### On entry,

A=&10

YX point to the address of the control block shown below :-

0	&84
1	0
2	Remote station (station number, network)
4	Pointer to start of local buffer for arguments
8	Pointer to end of local buffer for arguments
12	Procedure number
14	

In the remote machine the Accumulator (A register) holds the event number (which will be 8). X will hold the low byte of the procedure number and Y will hold the high byte.

### On exit,

A,X,Y undefined

0	Modified
1	Unchanged
14	

OSWORD A=&10 (Control byte=&84)

### Example

The first program uses the user procedure call to cause an event in a remote machine (held in the variable stn% : line 60). Note that the second program MUST be running in the remote station, otherwise the remote station will become protected.

```

10 REM Program to send Events to a remote machine
20 DIM blk% 100,buffer% 100
30 osword=&FFF1:osbyte=&FFF4
40
50 REPEAT
60 stn%=4 : cntr%=100
70 REPEAT
80 blk%?0=&84 : blk%?1=0 : blk%?2=stn%
90 blk%?4=buffer% : blk%?8=buffer%+4
100 blk%?12=2 :REM Procedure number
110 X%=blk% : Y%=blk% DIV 256 : A%=&10
120 CALL osword
130 cntr%=cntr%-1
140 UNTILFNpoll_transmit OR cntr%=0
150
160 UNTIL cntr%=0
170 PRINT"I can't send an event to machine ";stn%
180 END
190
200DEF FNpoll_transmit
210 LOCAL A%:A%=&32
220 REPEAT V%=(USR(osbyte) AND &FF00)
230 UNTIL (V% AND &8000)=0
240=((V% AND &7F00)=0)

```

This program is to be run in the remote machine. When an event occurs, the event routine increments a location on the screen.

```

10 REM Detecting an event sent through the Econet
20 REM (C) A.J.Engeham, SJ Research
30 REM
40 DIM code &100,buffer 100
50 osword=&FFF1:osbyte=&FFF4
60 MODE 7
70 PROCassemble
80 CALL set_up_event
90 PRINT"The event routine has been set up"
100 PRINT" - waiting for event to happen"
110 END
120
130DEF PROCassemble
140 FOR pass=0 TO 2 STEP 2
150 EVENTV=&220:zp=&70
160 P%=code
170[OPT pass
180.set_up_event
190 LDA EVENTV:STA zp \Save old event handling routine
200 LDA EVENTV+1:STA zp+1
210 LDA #handle_eco MOD 256

```

```

220 SEI                                \Changing Event vector interrupts
must
230                                \ be disabled.
240 STA EVENTV
250 LDA #handle_eco DIV 256:STA EVENTV+1
260 CLI                                \Vector done reenable interrupts
270 LDA #14:LDX #8:JSR osbyte          \Enable Econet events
280 RTS
290
300\***** Event handling routine *****
310.handle_eco
320 PHP
330 CMP #8:BNE not_econet_event        \If not 8 then the event was not
caused
340                                \ by the Econet
350 PHA:TXA:PHA:TYA:PHA                \It was the Econet so save registers
360
370 CPX #2                            \Check for User procedure call 2
380 BNE dud_eco_call                  \Not the procedure call sent from
390 CPY #0:BNE dud_eco_call            \ my machine
400 INC &7E00                          \Do something exciting
410.dud_eco_call
420 LDX #buffer MOD 256:LDY #buffer DIV 256 \YX points to the buffer
430 LDA #&12:JSR osword                \Read arguments
440 PLA:TAY:PLA:TAX:PLA                \Restore registers
450.not_econet_event
460 PLP                                \Restore flag
470 JMP (zp)                          \Pass event back to OS to handle
480 ]
490 NEXT
500ENDPROC

```

### Compatibility between Filing Systems :

Only supported by the NFS. The NFS need not be the current filing system.

## Remote Insert key

OSWORD A=&10 (Control byte=&85)

### General description

This call inserts a byte into a remote machine's keyboard buffer and is used by the 'send a line of text' command (OSWORD A=&14 Control byte=1), which should usually be used in preference to this call. See section 10.15 for information on polling the transmit for success.

### On entry,

A=&10

YX point to the address of the control block shown below :-

0	
1	&85
2	0
3	Remote station (station number, network)
4	Pointer to buffer
8	Pointer to buffer+1
12	
14	0

The buffer (which is 1 byte long) holds the byte to send to the remote station.

### On exit,

A,X,Y undefined

0	
1	Modified
2	Unchanged
14	

### Example :

```

10 REM Insert a character into a remote machine's keyboard buffer
20 REM (C) A.J.Engeham, SJ Research
30 REM
40
50 DIM blk% 16,buffer% 1
60 osword=&FFF1:osbyte=&FFF4
70 INPUT"Station number : ";stn%
80 PRINT"Character to send : ";
90 ?buffer%=GET
100 tries%=20

```

```

110 REPEAT
120 blk%?0=&85
130 blk%?1=0
140 blk%?2=stn%
150 blk%?4=buffer%
160 blk%?8=buffer%+1
170 blk%?12=0
180 X%=blk%:Y%=X% DIV 256
190 A%=&10:CALL osword
200 tries%=tries%-1
210 REPEAT
220 A%=&32
230 UNTIL (USRosbyte AND &8000)=0
240 tries%=tries%-1
250 UNTIL (USRosbyte AND &7F00)=0 OR tries%=0
260 IF tries%=0 PRINT"Unable to send character":END
270 PRINT"Character sent"
280 END

```

### Compatibility between Filing Systems :

Only supported by the NFS. The NFS need not be the current filing system.

## Start REMOTE

OSWORD A=&10 (Control byte=&85)

### General description

This call is used to start a REMOTE. See section 10.15 for information on polling the transmit for success.

### On entry,

A=&10

YX point to the address of the control block shown below :-

0	&85
1	0
2	Remote station (station number, network)
4	Pointer to buffer
8	Pointer to buffer+1
12	1
14	

### On exit,

A,X,Y undefined

0	Modified
1	Unchanged
14	

### Compatibility between Filing Systems :

Only supported by the NFS. The NFS need not be the current filing system.

## General description

This call causes a remote machine to write the state of its screen to a position in its Econet workspace (whence it can be read by the local machine by PEEKing this workspace). See section 10.15 for information on polling the transmit for success.

**On entry,**

A=&10

YX point to the address of the control block shown below :-

0	&85
1	0
2	Remote station (station number, network)
4	Pointer to buffer
8	Pointer to buffer+1
12	2
14	

**On exit,**

A,X,Y undefined

0	Modified
1	Unchanged
14	

In the remote machine the data is written to the address (in the I/O processor) pointed to by locations &9E and &9F plus &E9. This data is shown below :-

0	Address of top of screen
2	
18	Palette (physical colours defined on screen)
19	
21	Mode number (0-7)
23	
Mark	Address of start of screen
space for colours	

**OSWORD A=&10 (Control byte=&85)**

**Example :**

```

10 REM Info. about a remote m/c : Demonstration for OS procedure 2
20 REM (C) A.J.Engeham, SJ Research
30 REM
40
50 DIM blk% 100,buffer% 100
60 INPUT"Station number ",stn%
70 osword=&FFF1:osbyte=&FFF4
80
90 PROCtransmit(&85,1,2)
100 PROCtransmit(&81,2,&FFFF009E)
110 PROCtransmit(&81,30,&FFFF0000+(buffer%!0 AND &FFFF)+&E9)
120
130 MODE 7
140 PRINT"Machine number ";stn%
150 PRINT" MODE ";buffer%?18
160 PRINT" Address of start of screen="";~buffer%!19 AND &FFFF
170 PRINT" Address of top of screen="";~buffer%!0 AND &FFFF
180 PRINT'"Palette"' " " " " "Logical Physical"
190 FOR I%=0 TO 15
200 PRINTTAB(3);I%TAB(12);I%?(buffer%+2)
210 NEXT
220END
230
240DEF FNPoll transmit
250 LOCAL A%:A%=&32
260 REPEAT V%=(USR(osbyte) AND &FF00)
270 UNTIL (V% AND &8000)=0
280=((V% AND &7F00)=0)
290
300DEF PROCtransmit(cb%,no_of_bytes%,start%)
310 LOCAL cntr%,X%,Y%,A%
320 cntr%=20
330 REPEAT
340 blk%!0=cb%
350 blk%!2=stn%
360 blk%!4=buffer%
370 blk%!8=buffer%+no_of_bytes%
380 blk%!12=start%
390 X%=blk%:Y%=blk% DIV 256
400 A%=&10:CALL osword
410 cntr%=cntr%-1
420 UNTIL FNPoll transmit OR cntr%=0
430 IF cntr%=0 THENPRINT"Machine not listening":END
440ENDPROC

```

### Compatibility between Filing Systems :

Only supported by the NFS. The NFS need not be the current filing system.

## Fatal Error

OSWORD A=&10 (Control byte=&85)

### General description

This causes a fatal error in a remote machine. It is easier to use the high level fatal error call (OSWORD A=&14 Control byte=2). See section 10.15 for information on polling the transmit for success.

### On entry,

A=&10

YX point to the address of the control block shown below :-

0	&85
1	0
2	Remote station (station number, network)
4	Pointer to buffer
8	Pointer to buffer+1
12	3
14	

### On exit,

A,X,Y undefined

0	Modified
1	Unchanged
14	

### Compatibility between Filing Systems :

Only supported by the NFS. The NFS need not be the current filing system.

## Halt

OSWORD A=&10 (Control byte=&86)

### General description

Halts all non-interrupt operations in the I/O processor of a remote machine. If a tube is running on the remote machine, then that will continue running until it tries to communicate with the I/O processor. See section 10.15 for information on polling the transmit for success.

### On entry,

A=&10

YX point to the address of the control block shown below :-

0	&86
1	0
2	Remote station (station number, network)
4	

### On exit,

A,X,Y undefined

0	Modified
1	Unchanged
14	

### Compatibility between Filing Systems :

Only supported by the NFS. The NFS need not be the current filing system.

## Continue

OSWORD A=&10 (Control byte=&87)

### General description

Restarts the I/O processor of a remote machine after a Halt command (OSWORD A=&10 Control byte=&86). See section 10.15 for information on polling the transmit for success.

0	&87
1	0
2	Remote station (station number, network)
4	

### On exit,

A,X,Y undefined

0	Modified
1	Unchanged
14	

### Compatibility between Filing Systems :

Only supported by the NFS. The NFS need not be the current filing system.

## Identify Machine

OSWORD A=&10 (Control byte=&88)

### General description

This call interrogates a remote machine returning codes containing values to distinguish between manufacturers, machine types and software versions. See section 10.15 for information on polling the transmit for success.

### On entry,

A=&10

YX point to the address of the control block shown below :-

0	&88
1	0
2	Remote station (station number, network)
4	Pointer to start of local buffer
8	Pointer to end of local buffer (start+4)
12	

### On exit,

A,X,Y undefined

0	Modified
1	Unchanged
14	

The first four bytes of the buffer are relevant. The meaning of these bytes are shown below :-

### Byte 1

This is defined by manufacturers and is intended to indicate the hardware design of the machine. The following are currently defined :-

Value	Type of machine
<b>SJ Research</b>	
FF	Z80 CP/M
FE	SJ Research File Server
FD	RM 480Z
FC	Nascom 2 (running CP/M)
FB	IBM Interface board

FA SCSI Interface board

#### Acorn

01	BBC microcomputer (includes B+)
02	Atom
03	System 3 or System 4
04	System 5
05	Master
06	Electron
07	Reserved for future machine
08	Reserved for future machine
09	Communicator
0A	Econet Terminal
0B	File Store
0C	Compact

#### Byte 2

This byte indicates the manufacturer. The following are currently defined :-

Value	Manufacturer
00	Acorn
01	Torch
02	Reuters
FF	SJ Research

#### Bytes 3 & 4

These contain the low and high bytes of the software release version (in base 16) and are manufacturer specific.

#### Compatibility between Filing Systems :

Only supported by the NFS. The NFS need not be the current filing system.

## 10.10 Set up Rx block

OSWORD A=&11

### General description

This call is used to set up a receive block in your local machine. Once this has been done, data which is transmitted, or broadcast, on your port number, can be received.

### On entry

A=&11

YX point to the address of the control block shown below :-

0	0
1	&7F
2	Port number
3	Station number
5	Pointer to start of buffer
9	Pointer to end of buffer
13	

If the port number is zero then the receive block will receive data transmitted on any port number. If the station number is zero then the receive block will receive data sent from any station, as long as the port number is correct. Note that the size of the buffer must be big enough to receive all the data transmitted otherwise the data will be rejected, and the receive block left open. To select the correct port number for your application see the Econet Standards Group paper 0001 at the end of this section.

### On exit,

A,X,Y undefined

0	Receive block number
1	Unchanged
13	

If the receive block number allocated is zero then all the available receive blocks have been used. To correct this delete an unused receive block and retry the call.

## Read Rx block

OSWORD A=&11

### General description

Once data has been received into an open receive block a read operation is needed to discover the source of the data and the number of bytes actually received. This call allows a copy of the updated control block to be written to the address pointed to by YX. Reading a receive block deletes the old receive block and allows that receive block to be reallocated. If the receive block was open to all stations, by using a station number of 0, then the actual machine number that sent the packet will be written; this is also the case for the port number. See section 10.15 for details of polling to see when data has arrived.

### On entry,

A=&11

YX point to the control block shown below :-

0	Control block number
1	Undefined
13	

### On exit,

A,X,Y undefined

0	Control block number
1	Control byte of transmit packet from remote machine
2	Port number of remote machine
3	Station number of remote machine
5	Pointer to start of data
9	Pointer to end of data
13	

## 10.11 Read arguments

OSWORD A=&12

This call allows arguments to be read after a remote JSR (OSWORD A=&10 control byte=&83) and a user procedure call (OSWORD A=&10 control byte=&84). The remote machine will have the protect status bits set until this call is used, this is to prevent the arguments from being overwritten. The maximum size of the argument block is returned by OSWORD A=&13, function code=9.

### On entry,

A=&12

YX point to the address in memory to put the arguments, of which the first two bytes are the source station number.

### On exit,

A,X,Y undefined.

## 10.12 OSWORD A=&13

## Call Summary

## FS number

OSWORD A=&13 (control byte=0/1)

### General description

These calls allow general local station information to be read and written.

### On entry,

A=&13

YX point to the control block with the first byte determining the function and number of parameters. Except for function codes 0,1,6 and 7 the NFS does not need to be selected to perform the operations.

Control byte	Function
0	Read File Server number
1	Write File Server number
2	Read Printer Server number
3	Write Printer Server number
4	Read Protection mask
5	Write Protection mask
6	Read context handles
7	Write context handles
8	Read local station number
9	Read number, and size, of arguments
10	Read error number

### On exit,

A,X,Y undefined  
Control block modified

### General description

These calls allow the File Server number to be read and written. This value is used for all File Server commands including \*I AM <user identifier> and OSWORD A=&14. To change onto a different File Server, context handles need to be written otherwise it is likely that any further operations will return the error 'Channel'. The ANFS utility '\*FS' does both of these operations.

The BBC NFS ROM defaults to the value 0.254 (i.e. station 254 on the local network). On the Master, Compact and Econet Terminal the default File Server number is held in CMOS ROM and is configured by the command '\*CONFIGURE FS <new number>'.

These calls should only be attempted with Econet selected as the current filing system.

### On entry

A=&13

YX point to the control block shown below :-

0	Control byte 0 = Read FS number 1 = Write FS number
1	
2	File Server local number
3	File Server network number

### On exit,

A,X,Y undefined

For read operation the control block is modified as shown above. For write operation the control block is undefined.

## PS number

### General description

These calls concern the machine using a Print Server. To send a listing it is also necessary to set the printer type to the network, by issuing the command '\*FX5,4'. The BBC NFS ROM defaults to the value 0.235 (i.e. station 235 on the local network). On the Master, Compact and Econet Terminal the default File Server number is held in CMOS ROM and is configured by the command '\*CONFIGURE PS <new number>'.

### On entry,

A=&13

YX point to the control block shown below :-

0	Control byte 2 = Read PS number 3 = Write PS number
1	
2	Print Server local number
3	Print Server network number

### On exit,

A,X,Y undefined

For read operation the control block is modified as shown above. For write operation the control block is undefined.

OSWORD A=&13 (control byte=2/3)

## Protection mask

### General description

These calls allow the user to read and write the protection mask of the local machine. Note that if a remote machine attempts to do an operation requiring parameters returned then various protection bits will be set until the parameters have been read (using OSWORD A=&12).

### On entry,

A=&13

YX point to the control block shown below :-

0	Control byte 4 = Read protection mask 5 = Write protection mask
1	
2	Value of protection mask

If a bit of the protection mask is clear then the remote operation is allowed. The possible values of the protection mask are shown below :-

bit	Operation
0	Peek
1	Poke
2	JSR
3	User procedure call
4	Operation system procedure call
5	Halt

### On exit,

A,X,Y undefined

For read operation the control block is modified as shown above. For write operation the control block is undefined.

OSWORD A=&13 (control byte=4/5)

## Context handles

OSWORD A=&13 (control byte=6/7)

### General description

Three directory handles describing the current environment are required for most File Server operations. These are returned by the File Server by the command '\*I AM <user identifier>'. The three handles are :-

User root directory (URD) - starting directory for that user. If the command '\*DIR' is issued then the user will be returned to this directory.

Currently selected directory (CSD) - the directory that the user is currently in.

Library directory (LIB) - the directory that is searched if a filename is not found in the CSD.

These calls should only be attempted with Econet selected as the current filing system.

### On entry,

A=&13

YX point to the control block shown below :-

0	Control byte 6 = Read context handles 7 = Write context handles
1	URD
2	CSD
3	LIB
4	

### On exit,

A,X,Y undefined

For read operation the control block is modified as shown above. For write operation the control block is undefined.

## Read station

OSWORD A=&13 (control byte=8)

### General description

This call returns the local station number.

### On entry,

A=&13

YX point to the control block shown below :-

0	8=Read local station number
1	
2	undefined

### On exit,

A,X,Y undefined

Control block is modified to hold local station number.

### Example :

```
10 DIM blk% 2
20 PRINT FNstation
30 END
40
50DEF FNstation
60 A%=&13
70 X%=blk%:Y%=blk% DIV 256
80 blk%?0=8
90 CALL &FFF1
100=blk%?1
```

## Read arg. info.

OSWORD A=&13 (control byte=9)

### General description

This call is used to find the size of the argument block used by remote JSR and remote procedure call, currently held by the NFS, and the maximum size of argument block possible.

### On entry,

A=&13

YX point to the control block shown below :-

0	9=Read arguments
1	undefined
3	

### On exit,

A,X,Y undefined

Control block is modified as shown below :-

0	undefined
1	Number of arguments
2	Maximum number of arguments
3	

## Read error

OSWORD A=&13 (control byte=10)

### General description

This call is used to find the actual error number after a 'catch-all' error (error number &A8) has happened, or the screen MODE after a 'Mode x' error from '\*VIEW'.

### On entry,

A=&13

YX point to the control block shown below :-

0	10=Read errors number
1	undefined
2	

### On exit,

A,X,Y undefined

Control block is modified as shown below :-

0	undefined
1	Error number
2	

## 10.13 Send to FS

OSWORD A=&14 (control byte=0)

### General description

This call is used to communicate directly with the File Server and is detailed fully in the section detailing the File Server interface. The network must be the current filing system.

### On entry,

A=&14

YX point to the control block shown below :-

0	0=Communicate with File Server
1	Size of whole of block (x)
2	0 - NFS will put reply port here
3	FS Function code
4	0 - NFS will put URD handle here
5	0 - NFS will put CSD handle here
6	0 - NFS will put LIB handle here
7	Rest of FS transmit block
x	

### On exit,

A,X,Y undefined

Control block is modified as shown below :-

0	0
1	Size of rest of block
2	Command code
3	Return code
4	Rest of file server receive block
x	

## Send text

OSWORD A=&14 (control byte=1)

### General description

This call sends a message to a remote machine and is used by the library utility \*NOTIFY. The NFS must be the current filing system.

### On entry,

A=&14

YX point to the control block shown below :-

0	1=Send text
1	Destination station
3	Text, terminated by 0 or &0D
x	

### On exit,

A,X,Y undefined

Contents of control block is undefined.

The maximum size of text is 128 bytes when performed from a 2nd processor and 250 bytes when performed from the I/O processor. This call should not be used from within an interrupt routine.

## Cause remote error

OSWORD A=&14 (control byte=2)

### General description

This call causes a fatal error in a remote machine, by executing a BRK instruction followed by another 0, thus generating error 0. The NFS must be the current filing system. This is only guaranteed to be a fatal error in BASIC 2 and above. In BASIC 1 the ON ERROR will still trap fatal errors. Some other languages also incorrectly implement trapping of error 0.

### On entry,

A=&14

YX point to the control block shown below :-

0	
1	2=Cause fatal error
3	Destination station

### On exit,

A,X,Y undefined

Contents of control block is undefined.

## 10.14 OSBYTE

Entry point &FFF4  
Indirected via &20A

### On entry,

A=Function code

Value in A	Function
A=&32	Poll transmit block
A=&33	Poll receive block
A=&34	Delete a receive block
A=&35	Sever remote connection

### On exit,

X=Result dependent on the function

A,Y,C undefined

## Call Summary

## Poll Transmit

OSBYTE A=&32

### General description

This call is used to poll a transmit operation for completion. If an error occurred the error code will be returned in X.

### On entry,

A=&32

### On exit,

X=0 if the transmission has been completed successfully, otherwise the following bits in X are relevant. Bit 7 is the top bit, bit 0 is the bottom bit.

bit	state	Meaning
7	0	completed
	1	in progress
6	0	successful
	1	failed
5	0	
4	0	
3	0	
2-0		hold the error code

### Error codes

0	Line jammed.
1	Incomplete or bad four-way handshake
2	No acknowledge to scout packet
3	No clock
4	Bad transmit control block

A,Y,C undefined

### Example :

```
DEF FNpoll_Tx
  LOCAL Result%,A%
  A%=&32
  REPEAT
    Result%=(USR(osbyte) AND &FF00) DIV &100)
  UNTIL Result%<&80
  =Result%
```

The BASIC function above will poll the last transmit operation until it has been completed. When the function returns, if the result is non zero, then there has been an error in the transmission and the transmit command may have to be redone.

## Poll Receive

OSBYTE A=&33

### General description

This call is used to check whether anything has been received in a receive block.

### On entry,

A=&33

X=control block number

### On exit,

X has the top bit set if a message has been received.

A,Y undefined

C undefined

### Example :

```
DEF FNpoll_Rx(X%)
  LOCAL Result%,A%
  A%=&33
  =((USRosbyte AND &8000)<>0)
```

The above example, when given a control block number as a parameter, will return -1 (TRUE) if anything has been received.

```
10 CB%=FNset_up_Rx :REM Your function which sets up a receive block
20 :REM and returns the control block number.
30 REPEAT
40 UNTIL FNpoll_Rx(CB%)
50 PRINT"Something has been received!"
```

## Delete receive block

### General description

This call is used to delete a receive block.

### On entry,

A=&34

X=control block number

### On exit,

X control block number.

A,Y undefined

C undefined

OSBYTE A=&34

## 10.15 Transmitting

To transmit a packet over the network using the BBC microcomputer it is necessary to set up a *control block* in memory. A transmit control block will always take the form :-

0	Control byte (top bit set)
1	Port (must be zero for immediate operations)
2	Station number
4	Start of data
8	End of data
12	Remote start of data
16	

The *control byte* always has the top bit set. The rest of the byte is called a *control code* which, with the exception of immediate operations, makes no difference what value is chosen.

Check that the port number is within the correct range for your application by checking it against the Econet Standards Group paper 0001.

### General procedure for transmitting

- 1) Set up the control block and the number of retries to perform.
- 2) Store the control byte in the control block.
- 3) Set the accumulator to &10 and YX to point to the control block.
- 4) CALL OSWORD to perform the call.
- 5) Read the control byte of the control block. If this is zero then the transmission has failed to start so go to step 2.
- 6) Poll for completion (Using OSBYTE A=&32).
- 7) If not completed (i.e. top bit of X register still set) go to step 6.
- 8) Has transmission worked (Using OSBYTE A=&32 to find whether X is 0).
- 9) If there was a non-fatal error in the transmission then decrement the number of retries and if the number of retries is greater than 0 go to step 2.
- 10) If the error was fatal or the number of retries=0 then finish otherwise return.

The terms *fatal* and *non-fatal* refer to the type of error returned by OSBYTE with A=&32. The errors 'line

jammed' (&40), 'no clock' (&43) and 'bad transmit control block' (&44) are all fatal: this is because they all need manual intervention by the user to cure them, a detailed explanation of these errors is given in appendix A.

The errors 'four way handshake damaged' (&41) and 'no scout acknowledgement' (&42) are termed non-fatal. The error 'no scout acknowledgement' is normally caused by either, the remote machine having no receive block open to receive the data, the machine has the protection status set or the machine is not on the network. The error 'four way handshake damaged' is normally caused by the receive buffer size of the remote machine not being big enough, the data colliding with someone else transmitting or electrical interference. It should be noted that the Acom BBC B+ and early Master series microcomputers do not have any collision detect hardware (although it can be fitted) and so it is possible that it may not always transmit correctly, especially on the broadcast operation.

## Broadcasting

This is a special version of the transmit operation. It allows a station to simultaneously send eight bytes of data to every other station on every network, that has a receive block open and is currently the only method of communicating with the *bridge*. However, the call has no handshaking so it is not possible to guarantee that the remote stations received the data. The control block is shown below :-

0	Control byte (top bit set)
1	
2	Port
4	&FFFF
12	Data

If the remote station has a receive block open on the correct port number then the data will be received like a normal transmit, except that if the remote station has NFS version 3.34 the data will be written over the pointers of the receive control block instead of the location pointed to.

## Transmitting under interrupts

If an application requires performing a transmission under interrupts then there is a special precaution to take. This is because the transmit routine interrupts and so there is the possibility that a transmission under interrupts may occur whilst a transmission is already taking place. Thus polling the transmit will get the result of the last transmission. To get around this problem the following *fix* must be done :-

Before transmitting save the old transmission poll flag by :-

```
loop
  LDY #&6F
  LDA (&9C),Y
  BMI loop
  PHA
```

Having successfully transmitted, restore the old transmission poll flag by :-

```
LDY #&6F
PLA
STA (&9C),Y
```

It is essential that the above routines are executed in the I/O processor.

## 10.16 Receiving

A receive block is needed to receive a transmission from a remote machine. The only exceptions to this are the immediate operations, these are handled specially by the NFS ROM.

Before a receive block can be used it must be set up. This is performed by setting up a control block and CALLing OSWORD A=&11. If a valid receive block number is returned then the receive block is open. The receive block will only be filled if all the following conditions are met :-

- 1) The correct station replies
- 2) The data is sent on the correct port number
- 3) The amount of data sent is not bigger than the buffer size reserved.

## 10.17 Port numbers

The information below is a copy of a paper by the Econet Standards Group.

### ESG paper 0001

This standard defines the way in which port number may legally be used. They should only be used for the purposes given in the following list.

00	Illegal (note 1)
01-0F	Reply only (note 2)
10-2F	Torch Computers (note 3)
30-8F	Reserved for future allocation
90-9F	Acom Computers (note 3)
A0-AF	SJ Research (note 3)
B0-CF	Reserved for future allocation
D0-DF	Acom Computers (note 3)
E0-FE	User Programs
FF	Illegal

### Notes

1) Used to provide immediate operations.

2) These ports may be used by anyone for any purpose, subject to certain restrictions. They may not be used to initiate a new protocol, but may be used to receive replies within a protocol initiated using some other port. When receiving data on these ports, the receive block must be open for a specific station, and not for all stations. Data should never be sent on these ports unless it is certain that the recipient is prepared for that data, and is carrying out the same protocol. Broadcasts may not be sent on these ports.

These ports must only be used by software which has complete control of one end of the connection, and can ensure that no other software will attempt to transmit or receive on these ports at the same time. It is suggested that unsolicited traffic on these ports be discarded immediately by any software that receives it.

3) Where numbers are allocated to a particular organisation, that organisation will control the use of those numbers. However, it is to be hoped that, as new standards are agreed, some or all of these numbers will be allocated to specific purposes at a later date.

### Currently used ports

Ports which are currently used are :-

99	Command port; used for sending a command to the file server.
9E	Printer server status reply port; used by the printer server to reply to a client communicating on port 9F.
9F	Printer server status port; used to ask for current status from the printer server.
A0	Used by the SJ Research communication protocol 'FAST'.
D1	Printer server data port; used for communicating to the print server.

## 10.18 The Bridge

The number of stations on any one network is limited to 254, because the station number is a one byte number and values 255 and 0 are reserved for special purposes. An Acorn bridge is used to connect two Econet networks together; bridges can be used to connect up to 127 networks together, giving a maximum of 32258 stations in total.

Another use of a bridge is to split a long network so that the data transfer rate inside a network can be fast. This is because each network will be shorter; only when data is being transferred between networks does the data transfer rate slow down. If a fault appears on one of the networks then the other network will continue to work; this is particularly useful for developing and testing new Econet hardware.

The last use of a bridge is to allow otherwise illegal network layouts, the most common of these being a 'T' junction.

### How to use the bridge

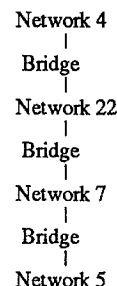
All the primitives supplied by the NFS ROM have two bytes reserved for the station number. The first byte is the station number and the second byte is the network number. To communicate with machines on the local network the network number should be 0.

For a station on network 4 to communicate with another station on network 4, the network number would be 0. But, for a station on network 5 to communicate with a station on network 4, the network number would be 4.

When the bridge is not transferring data, it looks at all the scout packets being sent on both of the networks that it is bridging across, called the networks *adjacent* to the bridge. If the scout packet contains a network number which is within reach through the bridge, then the bridge will perform a four-way handshake between the networks to transfer the data.

To do a four way handshake the bridge receives data from one adjacent network and puts it into its own RAM. While this is done, the bridge sends *flags* on the other adjacent line to prevent any network activity. Once the transfer of data has been completed, the bridge moves to flagging the first network instead, while it transfers the data from its internal RAM onto the second network. Because the bridge has to wait until it has received all the data before re-transmitting it, the transfer rate halves when data is transferred across a bridge. Note that data being transferred between networks 4 and 5, in the system shown overleaf, would be approximately one quarter of the speed of a transfer inside network 4 (the actual figure depends on the clock speeds for each network).

When a bridge is switched on, it only knows about its adjacent networks. So it *broadcasts* a reset packet to find out what other networks are available. The other bridges on the network all update their network lists.



The bridge can be interrogated to find the number of the local networks, version number of the software running in the bridge and the numbers of any other bridges on the network. In addition to these calls the bridge also responds to a reset message and a normal data transfer message.

As the bridge does not use immediate operations, for transmissions, it is necessary to set up a receive block, to receive status information. Thus the process for communicating with the bridge is :-

- 1) Set up a receive block.
- 2) Broadcast the data to bridge.
- 3) Poll receive block until packet received or a timeout occurs.
- 4) If the polling time was excessive delete the receive block and finish.
- 5) Read the receive block and return results.

Normally the user will never need to communicate with the bridge as all necessary communications are handled by the network filing system in the computer. ANFS asks the bridge which network number it is on when the machine is first powered on. This is so that the user can type \*I AM 1.254 USER when he is on network 1. ANFS knows that the current network number is 1 so before sending the packet it changes the network number to 0.

There are two useful messages that can be sent to the bridge. These are the 'what net' and the 'is net' calls. The 'what net' asks the bridge what the number of the local network is. The packet to broadcast is :-

0	&82
1	&9C
2	&FFFF
4	"BRIDGE",reply port,0
12	

The remote bridge's reply packet will be 2 bytes long containing the local network number and the version of the software running in the bridge.

The other useful message that can be sent to a bridge is the 'is net' packet. This call is used to ask the bridge if it knows about network N, the bridge will perform a four-way handshake if the bridge number is

held in its internal table otherwise the bridge will not reply.

The broadcast packet is :-

0	
1	&83
2	&9C
4	&FFFF
12	"BRIDGE",reply port,N

The most obvious use of the above call is to find out the numbers of all other networks available. The BASIC program below does just this :-

```

10 REM Find out what other networks are available
20 REM (C) 1986, A.J.Engeham, SJ Research
30 REM
40
50 DIM blk% 100,tx_blk% 20,buffer% 100
60
70 osword=&FFF1:osbyte=&FFF4
80 reply_port=&8A:retries=10
90 PROCset_up_rx
100
110 FOR bridge%=1 TO 127
120 FOR dummy%=1 TO 3
130 PROCbroadcast(bridge%)
140 NEXT
150
160 dummy%=0
170 REPEAT
180 dummy%=dummy%+1
190 UNTIL dummy%=retries OR FNpoll_rx
200
210 IF FNpoll_rx THENPROCdisplay(bridge%)
220 NEXT
230 PROCdelete
240 PRINT"Finished"
250END
260
270DEF PROCset_up_rx
280 blk%?0=0 :REM control block number put here
290 blk%?1=&7F
300 blk%?2=reply_port
310 blk%?3=&0 :REM Receive from any stations
320 blk%?5=buffer%
330 blk%?9=buffer%+8
340 X%=blk%:Y%=blk% DIV 256
350 A%=&11
360 CALL osword

```

```

370 IF blk%?0=0 THENPRINT"Can't open a Rx block":END
380ENDPROC
390
400DEF FNpoll_rx
410 A%=&33
420 X%=blk%?0 :REM Get the rx block number
430=((USRosbyte) AND &8000)<>0
440
450DEF PROCbroadcast(network%)
460 LOCAL redo
470 redo=5
480
490 REPEAT
500 tx_blk%?0=&83
510 tx_blk%?1=&9C :REM Broadcast on port &9C
520 tx_blk%?2=&FFFF :REM Broadcast operation
530 $(tx_blk%+4)="BRIDGE"
540 tx_blk%?10=reply_port
550 tx_blk%?11=network%
560 X%=tx_blk%:Y%=tx_blk% DIV 256
570 A%=&10:CALL osword
580 A%=&32
590 REM Wait for completion
600 X%=FNpoll_tx
610 redo=redo-1
620 UNTIL redo=0 OR X%=&40 OR X%=&43 OR X%=&44 OR X%=0
630 IF X%>0 THEN PRINT"Broadcast failed, error "&X%:PROCdelete:END
640ENDPROC
650
660DEF PROCdelete
670 REM delete a receive block
680 A%=&34
690 X%=blk%?0
700 CALL osbyte
710ENDPROC
720
730DEF FNpoll_tx
740 REPEAT UNTIL NOT ((USRosbyte) AND &8000)
750=((USRosbyte) AND &FF00) DIV &100)
760
770DEF PROCdisplay(no%)
780 PRINT"Found network ";no%
790 PROCdelete
800 PROCset_up_rx
810ENDPROC

```

## 10.19 Printers

The protocol used by SJ Research to locate named printer resources is an adaptation of the protocol used by Acorn (and implemented in ANFS). The extra complexity is introduced by the possibility of having multiple printers at the same station, such that the printer server has to keep track of the current printer selection for each user. The printer status protocol is used : a status request is broadcast, containing the name of the printer desired, and the station, making the broadcast can chose which of the responding servers to select.

Data for printing is sent to the print server in packets, using port &D1. Each packet has a maximum length of 80 bytes. All packets received by the print server (including logon/ logoff packets) are acknowledged by sending a packet back to the client, when the server is ready to receive the next data packet. These ack packets are sent on port &D1, and are 1 byte long (this byte contains no information).

All packets have a sequence number in bit 0 of the control code. The purpose of this is to discard duplicate packets, which occur due to a fundamental problem with networks. It is possible for a packet to be sent and received correctly, but the transmitter failed (and so tries again) while the receiver is unaware of the problem and receives both copies as good packets. The solution is to attach a sequence number to each packet, and when two packets are received with the same sequence number, the second can be discarded. It turns out that a single bit is the control code : when repeating packets after a failed transmit, the sequence numbers should be kept the same. Ack packets are sent with the same sequence number as the data packet being acknowledged : when waiting for an ack packet, any that arrive with a different sequence number from the last data packet should be ignored.

A client logs on to the print server by sending a data packet to the print server with the control code = &82. This packet should contain one byte, value zero (although there is a bug in Acorn NFS whereby a number of ctrl-C characters may be sent in front of the zero). Further packets follow, containing the data to be printed, with a control code = &80+sequence. The client logs off by sending a packet with control code = &84+sequence bit. This packet may contain print data, but the last byte is discarded (it is assumed to be the ctrl-C from a BBC micro).

### Problems with printing

With NFS 3.34 printing anything other than straight text is not guaranteed. This is because it is not possible to send the characters 2 and 3 to the printer without them being interpreted specially, even by using VDU 1. There is another problem with the NFS inserting extra 'null' characters into the print stream.

With NFS 3.6 a new print protocol has been devised which allows any character to be sent to the Print Server, thus allowing graphics dumps to be printed.

When using the Econet Print Server it is necessary to understand the concept of logging-on and -off (The same as with an FS). Basically VDU 2 logs on, and VDU 3 logs off: on non-spooling Print Servers (eg the Acorn PS ROM, some SJ File Servers) logging-on has the effect of 'locking-out' other users so that they get a 'Not listening' error. Logging-on also has the effect of printing a banner, including possibly time of printing and the user's name, whilst log on & off flushes the print buffer and sends a form-feed. Therefore these two commands should not be used to enable/disable output to a printer (when some characters must go to screen but not the printer), even though this works OK with a local printer.

To enable/disable printer output after a VDU 2, use \*FX3 4 to disable the printer and \*FX 3 0 to enable it again. See page 442 of the BBC User Guide.

## 10.20 File Server Interface

### Introduction

This section deals with the interfaces through which clients (station users and user programs) can interact with the File Server. The interfaces allow a client to manipulate the filestore, to gather information about files, directories, other users of the file server and about the configuration of the file server.

The BBC microcomputer uses the fileserver interface for all filing system operations, including '\*\* commands, LOAD and SAVE. When a command is sent from the BBC microcomputer to the file server, it will first be converted to transmit packets. For example if the line \*I AM 0.235 TONY was typed, this would first set my fileserver number to 235 on my local network. It would then convert the string to a fileserver packet with a function code of 0 (OSCLI command) and send the packet using a control byte of &80 on port &90. The file server expects all commands to be sent to it on port &90, the network filing system ROM in the BBC will specify port &99 for the file servers to reply.

Once the BBC microcomputer has finished sending the data, the fileserver will send a reply packet. For a logon packet this will contain three context handles and a boot option. The NFS ROM in the BBC will convert these to its own internal format.

### Conventions and Abbreviations

All numbers used are given in decimal except where prefixed by '&' to indicate hexadecimal notation. CR is used as an abbreviation for 'carriage return', character &0D.

### The User Environment

A client is identified and authenticated to the file server by its station number and three HANDLES. These handles are created by the file server by opening directories, and identify to the file server the environment in which to interpret commands and to look-up filenames presented by the client. The handles are created by the file server when a user logs on and are closed when the user logs off again. The three handles which comprise the user environment are:

The Currently Selected Directory ( CSD ) - the directory in which to look up all filenames which do not refer explicitly to another directory.

The User Root Directory ( URD ) - the directory where a user is placed after logging-on and where he is returned after a \*DIR command. On SJ Research fileservers, the character '&' can be used to specify the URD within commands.

The Library ( LIB ) - the directory where unrecognised commands are looked up if the filename is not found in the CSD.

Usually the client machine software deals with the manipulation of these handles; but it is possible for a user to define his own environment by opening several directories, and declaring a set of these handles as representing the current environment. This enables the user to execute a command in a number of different environments.

Almost all file server commands need these three handles to be transmitted as part of the information.

### File Server Function Codes

These calls will be used for all communication with the file server. If these calls are made directly to the file server, in particular \*DIR and \*I AM, then it is possible that the software will not work with all versions of the NFS ROM, in particular the ANFS, because the filing system ROM will not be able to update its workspace. The control block outlined below is always sent to your currently selected fileservers; to change your fileserver number use the OSWORD call with A=&13, reason code=1.

Listed under each File Server call is the packet to be sent to the fileservers. The most convenient way to send packets to the fileservers is by use of OSWORD A=&14, which automatically inserts context handles and controls the packet transmission/reception. To use this call, a standard header must be added to the other information, as shown below.

## On Entry,

A=&14

The NFS must be the current filing system.

YX point to the address of the transmit control block, which is shown below :-

0	0
1	Size of whole block (n)
2	0
3	Function code
4	0
7	Rest of File server transmit block
n	

When the File Server call is made, the NFS ROM modifies the control block, as shown below, before sending it to the file server. This allows the user to communicate with the File Server without having to know his context handles, or which port he should communicate with the File Server on.

Bytes 0 to 7 are referenced throughout this section as the *transmit block*. Bytes 2 to n are the bytes that are actually sent through the network, to the File Server.

0	0
1	Size of whole block (n)
2	Reply port
3	Function code
4	User's Root Directory context handle in File Server format
5	Currently Selected Directory context handle in File Server format
6	Currently Selected Library context handle in File Server format
7	Rest of File server transmit block
n	

## On Exit,

A,X,Y undefined

The transmit control block is replaced by the receive control block shown below :-

0	0
1	Size of whole block (new value)
2	Command Code
3	Return Code
4	Rest of File server Receive block
x	

Bytes 0 to 4 are referenced throughout this section as the *receive block*. Bytes 2 to 4 are the bytes that are actually returned by the file server.

The *command code* indicates to the client what action (if any) the client should take upon receiving this response.

The *return code* is an indication to the client of any error status which has arisen, as a result of attempting to execute the command. A return code of zero indicates that the command step completed successfully; otherwise the return code is the error number indicating what error has occurred. If the return code is non-zero, then the remainder of the message contains an ASCII string terminated by a carriage return, which describes the error.

## Summary of File Server Calls

Function code Description

0	Command line decoding
1	Save
2	Load
3	Examine
4	Catalogue header (Acom only)
5	Load as command
6	Open file
7	Close file
8	Get byte
9	Put byte
10	Get bytes
11	Put bytes
12	Read random access information
13	Set random access information
14	Read disc name information
15	Read logged on users
16	Read date/time
17	Read EOF (end of file) information
18	Read object information
19	Set object information
20	Delete object
21	Read user environment
22	Set user's boot option
23	Logoff
24	Read user information
25	Read file server version number
26	Read file server free space
27	Create directory, specifying size
28	Set time/date
29	Create file of specified size
30	Read user free Space (Acom only)
31	Set user free Space (Acom only)
32	Read client user identifier
64	Read account information (SJ Research only)
65	Read/write system information (SJ Research only)

## Command Line Decoding

Function code=0

### General description

A number of the operations performed by the file server are initiated by the sending of a command line. The function code of zero indicates to the file server that such a command line has been received. All command line type exchanges have the same format:

Client (command port):

0	Transmit block (shown on summary page)
7	Command line terminated by a CR
n	

File Server (reply port):

0	Receive block (shown on summary page)
4	Command dependent results
n	

If the command requires more action by the client, then the command code indicates what command line the file server has decoded. The file server will also return any decoded parameters or data which the client will need to complete the command. The possible command codes that the file server may return are:-

0	No Action, command complete
1	*Save
2	*Load
3	*Cat
4	*Info, *Printer, *Printout
5	*I AM
6	*Sdisc (Acom only)
7	*Dir, *Sdisc (SJ Research only)
8	Unrecognised command
9	*Lib

## Description of returned command codes

The following receive blocks are the result of command line decoding done by the file server. Any context handles returned by the file server are in file server internal format.

### \*SAVE Command code 1

0	Receive block (shown on summary page)
4	File load address
8	File execute address
12	File size
15	File name terminated by a CR
n	

The protocol then continues with function code 1

### \*LOAD Command code 2

0	Receive block (shown on summary page)
4	File load address
8	Flag : If flag=&FF then load address is defined
9	File name terminated by a CR
n	

The protocol then continues with function code 2.

### \*CAT Command code 3

0	Receive block (shown on summary page)
4	Decoded directory name terminated by CR
n	

### \*INFO Command code 4

0	Receive block (shown on summary page)
4	Character string of information terminated by negative byte (&80)
n	

### \*I AM Command code 5

0	Receive block (shown on summary page)
4	URD handle
5	CSD handle
6	LIB handle
7	Boot option (4 least significant bits)
8	

### \*SDISC Command code 6

0	Receive block (shown on summary page)
4	URD handle
5	CSD handle
6	LIB handle
7	

### \*DIR Command code 7

0	Receive block (shown on summary page)
4	CSD handle
5	

Unrecognised command

Command code 8

0	Receive block (shown on summary page)
4	
n	Command string terminated by CR

\*LIB

Command code 9

0	Receive block (shown on summary page)
4	
5	LIB handle

Note that CSD, URD and LIB handles returned by the File Server are not suitable for selecting with OSWORD A=&13 as the NFS maps these onto a set of internal handles. Hence these commands should usually be issued via the operating system.

## SAVE

Function code=1

### General description

This call requests to save a file. This call will be made following a \*SAVE command line sent to the file server (used by the Acorn SYSTEMs and Atoms). Note that the BBC computer interprets the parameters to a \*SAVE command internally and will enter the protocol by issuing a save with function code 1.

Client (command port):

0	Standard Transmit block except data acknowledge port replaces URD handle
7	
11	file load address
15	file execute address
18	file size
n	file name terminated by CR

File server (reply port):

0	Standard Rx Header
4	
5	data port
7	maximum data block size
n	file name terminated by CR (only present if command code is 3)

The client and file server now enter the 'data exchange phase' of the protocol, where the file server acknowledges the receipt of each data packet. If there is no data to be sent (eg a zero length file) then this phase is omitted. If the file server detects an error during the data transfer phase (eg a disc error), then the 'data exchange phase' is allowed to complete but the SAVE operation is aborted. The error status is returned in the return code of the 'final acknowledge'. Because the data is not transferred on the command port the following data must be sent by direct transmissions to the file server.

Client (data port):

0	A block of data, up to the maximum data block size.
n	

File server (data acknowledge port):

0	Undefined
1	

When the final data block has been received by the file server, the 'data acknowledge' is replaced by the 'final acknowledge', which is the terminating packet of the protocol.

File server (reply port):

0	Command code
1	Return code
2	Access byte (in standard format)
3	file creation date
5	

## LOAD

Function code=2

### General description

This call requests to load a file. This call will be made following a \*LOAD command line sent to the file server (used by the Acom SYSTEMs and Atoms). Note that the BBC computer interprets the parameters to a \*LOAD command internally and will enter the protocol by issuing a load with function code 2. This call will not work via OSWORD A=&14.

Client (command port):

0	Standard Transmit block with byte 4 containing dataport
7	
n	File name terminated by CR

File server (reply port):

0	Standard Rx Header
4	file load address
8	file execute address
12	file size
15	file access (as for SAVE)
16	file creation date (as for SAVE)
18	file name terminated by CR
n	

If the file is of zero length then the 'data transfer' phase is omitted. If the file server detects an error (eg disc error), then the required amount of data will be sent but its data content is undefined. Because the data is not transferred on the command port the following data must be sent, and received by direct transmission and reception.

Data is transmitted until 'file size' data has been sent.

File server (data port):

0	Data block
n	

File server (reply port):

0	Command code
1	Return code
2	

## Examine

Function code=3

### General description

This call returns information about files in a particular directory. The detail of the information is dependent on the value of ARG.

- 0 - All information, machine readable format
- 1 - All information, character string
- 2 - File title only
- 3 - File title and access, character string

Client (command port):

0	Transmit block (shown on summary page)
7	
8	ARG
9	Entry pointer to directory
10	Number of objects to examine
n	Directory name (terminated by a <CR>)

The directory entry pointer gives the entry number within the directory from which to examine. Conventionally the first entry in a directory is entry number zero.

The number of entries to examine specifies how many entries are to be examined, so is usually determined by the buffer space available to the client. A parameter of zero in this case conventionally demands that all entries in the directory from the entry point to the end of the directory be examined. On an SJ Research File Server if 0 entries are requested then no entries are returned.

Where information is returned in character string format, individual entries are delimited by zero bytes (&00), the final entry being terminated by a negative byte (&80). Carriage returns may occur within such strings. Where data is returned in machine readable format, the entries consist of a defined number of bytes; and so there are no delimiters between entries although the final entry is terminated by a negative byte.

# On exit,

If ARG=0

0	Receive block (shown on summary page)
4	No. of entries examined
5	Cycle number
6	Object's title (Padded with spaces)
16	Reload address
20	Execution address
24	Access (bits 7-0: M P D L W R w r)
25	Date object last updated
26	Year & month object last updated (top 4 bits = year since 1981, bottom 4 bits = month)
27	Main account number (low byte)
28	Top 3 bits are main account number Bottom 3 bits are aux. account number
29	Aux. account number (lo byte)
30	Size of file (in bytes)
33	Next file (as above, bytes 5-31)
n	

Note that the Acorn file server will return its system name in bytes 27 to 30. If an entry in the print queue has been examined then the access D/R is a /spl file and an access D/WR is a /prt file.

If ARG=1,

0	Receive block (shown on summary page)
4	No. of entries examined
5	Cycle number
6	Character string of all information
n	

If ARG=2

0	Receive block (shown on summary page)
4	10 (object name length for BBC MOS)
5	object name padded with spaces
n	

If ARG=3

0	Receive block (shown on summary page)
4	object name padded to 10 characters followed by formatted access string
22	

Data terminated by an &80.

## Catalogue Header

### General description

This call returns the data used in the \*CAT command.

### On entry,

0	Transmit block (shown on summary page)
7	
n	Directory name (CR returns information for CSD)

### On exit,

0	Receive block (shown on summary page)
4	
14	Last component of directory name padded with spaces
15	Character indicating ownership of directory
18	three space characters (&20)
33	Current disc name padded with spaces (Terminated by CR, negative byte)

Note that this call is only fully supported by Acorn File Servers as it is only used by Systems 3/4/5 and Atoms.

## Function code=4

## Load as Command

### General description

### On entry,

0	Transmit block (shown on summary page)
7	
n	file name (Terminated by CR)

The remainder of this protocol is exactly as for function code 2 (LOAD), except the file name is looked up first in the CSD, and if not found then, also in LIB. It is used for loaded '\*\*' commands. The error returned if the file name is not found in either directory is 'Bad command'. On an SJ Research File Server the call is equivalent to function code=5 except for run only users.

## Function Code=5

## Find (OPEN)

### General description

This function code creates a handle for the object specified, with the access type requested. Such handles are used for performing random access operations and also for manipulating the user's environment. An object will be opened only if the client has the necessary access rights to the object. A file can be opened several times for reading only, but only once for update. A file will be created if:

- 1: The file does not exist
- 2: The file is opened for update

### On entry,

0	Transmit block (shown on summary page)
7	
8	
9	
n	object name terminated by CR

### On exit,

0	Receive block (shown on summary page)
4	
5	file handle

Function Code=6

## Close

### General description

This function indicates to the file server that the handle passed as argument is no longer needed, and that all of the updated data in the file should be written out to the disc. A handle of zero indicates to the file server that all handles to open FILES are to be closed. This call does not close handles to directories. Note that the handle is a File Server format handle.

### On entry,

0	Receive block (shown on summary page)
4	
5	Handle

### On exit,

0	Receive block (shown on summary page)
4	

Function Code=7

## BGET

Function Code=8

### General description

The next four function codes deal with the facilities that the file server provides to enable the user to perform random access operations on open files. These operations have an additional protocol to ensure the integrity of the data exchanged, provided by a SEQUENCE NUMBER. The sequence number is a single bit, held in both client station and file server, which differentiates between successive reads of a file using the pointer held in the file server, and repeated reads of the same byte because the operation failed at the first attempt. The sequence number is sent in the least significant bit of the flag byte of the Econet control block. The file server will return its copy of the sequence number with the data to allow the client to detect data sequencing errors.

The client should invert his copy of the sequence number after every successful transaction with the file server. If the client detects a data packet with the incorrect sequence number then the client should be prepared to repeat the request.

This function code reads a byte from the file at the position specified by the file server's internal file pointer. Note that the handle is a File Server format handle.

### On entry,

0	0
1	Size of whole block (5)
2	Reply port
3	Function code (8)
4	File handle
5	

### On exit,

0	Receive block (shown on summary page)
4	byte read &FE if reading first byte after file end
5	flag byte &00 = normal read operation &80 = last byte in the file &C0 = first byte after file end
6	

## BPUT

Function Code=9

### General description

This function code writes a single byte to the file at the position indicated by the file server's internal file pointer. Note that the handle is a File Server format handle.

### On entry,

0	0
1	Size of control block (6)
2	Reply port
3	Function code (9)
4	File handle
5	Byte to be written
6	

### On exit,

0	
4	Receive block (shown on summary page)

# Getbytes and Putbytes

## General description

These operations allow the client to access blocks of data. The client may supply an offset within the file at which to start the operation, or may use the sequential file pointer maintained by the file server. The protocol includes a sequence number as described for BGET and BPUT.

### On entry,

0	0
1	Size of rest of block
2	0
3	Function Code
4	Data Ack Port
5	CSD handle
6	LIB handle
7	File Handle
8	non-zero = use FS sequential pointer Zero = use supplied offset
9	Number of bytes
12	File offset (if supplied)
15	

Reply to the first call is :-

0	Receive block (shown on summary page)
4	Data port (function code 11 only)
5	Maximum data block size (function code 11 only)
7	

The data transfer phase is exactly as described for LOAD and SAVE. For transfers of zero data these steps are not executed. If a read extends over the end of the file then the requested amount of data will be returned, but the number of valid data bytes will be less than the amount of data requested. The remaining data is undefined.

### Function Code=10 and 11

### On exit,

0	Receive block (shown on summary page)
4	0 = all ok &80 = read includes last byte in file (Value undefined for PUTBYTES)
5	Number of valid data bytes transferred
8	

## Read Random Access Information<sup>Function Code=12</sup>

### General description

This function code allows the client to discover information about files for which he currently has handles.

### On entry,

0	Transmit block (shown on summary page)
7	
8	
8	File handle
9	0 = sequential file pointer 1 = file extent (the amount of valid data) 2 = file size (the space allocated for the file)

### On exit,

0	Receive block (shown on summary page)
4	
7	
7	Information requested (low byte first)

## Set Random Access Information<sup>Function Code=13</sup>

### On entry,

0	Transmit block (shown on summary page)
7	
8	
8	File handle
9	0 = set sequential file pointer 1 = set file extent
12	Value to set (low byte first)

### On exit,

0	Receive block (shown on summary page)
4	
4	

Setting the extent of a file is not supported on some early File Servers.

## Read Disc Information

Function Code=14

### General description

This function returns the disc configuration of the file server. Conventionally the first drive in the file server has drive number zero. If the number of drives to interrogate is zero then the file server will return information on all drives in the system. It is not necessary to be logged on to read this information.

### On entry,

0	Transmit block (shown on summary page)
7	First drive number
8	Number of drives to interrogate
9	

### On exit,

0	Receive block (shown on summary page)
4	Number of drives found
5	Drive number of first drive selected
6	Disc name padded with spaces
22	Drive number of second drive selected
23	.
	.
n	.

## Read Current Users

Function Code=15

### General description

This function returns the currently logged-on users of the file server, their station numbers and associated privileges. Conventionally the logged-on user entries start at zero. A request for zero entries will return information for all users, commencing at the start entry.

### On entry,

0	Transmit block (shown on summary page)
7	Start entry
8	Number of entries to get
9	

### On exit,

0	Receive block (shown on summary page)
4	Number of entries returned
5	Machine number of first user (low byte, high byte)
7	First user name terminated by CR
n	0 = User not privileged non-zero = user privileged
(n+1)	Second user name terminated by CR
	.
	.
	.

## Read Date and Time

### General description

It is not necessary to be logged-on to the file server to use this function code.

#### On entry,

0	Transmit block (shown on summary page)
7	

#### On exit,

0	Receive block (shown on summary page)	
4		
Date		
1st byte days		
2nd byte (top 4 bits) - years since 1981		
(low 4 bits) - month		
6	Time	
1st byte hours		
2nd byte minutes		
3rd byte seconds		
9		

If the call is performed on an Acom level 2 File Server without a time board then the three bytes containing the time will be set to 0. Beware that this software can misinterpret this result by reading the time at midnight and deducing that no time board is attached!

Function Code=16

## Read 'End-of-file' status

### General description

This function is valid for file handles only.

#### On entry,

0	Transmit block (shown on summary page)
7	
8	File handle

#### On exit,

0	Receive block (shown on summary page)
4	
5	0 = file pointer within file &FF = file pointer outside file

Function code=17

# Read Object Information

Function Code=18

## General description

This call returns detailed information about a file or directory. The File Server call with ARG=64 is only supported on the SJ Research file server.

## On entry,

Client (command port):

0	Transmit block (shown on summary page)
7	
8	Object name terminated by CR
n	

## On exit,

The results returned depend on the argument passed with the call.

ARGs 1-5

File server (reply port):

0	Receive block (shown on summary page)
4	
5	
	Results returned (load/exec/size/access/date/ownership) ownership = &00 -> owner = &FF -> public
n	

ARG 6

File server (reply port):

0	Receive block (shown on summary page)
4	
5	
7	Directory name padded with spaces
17	
	Access to directory &00 = owner access &FF = public access
18	
19	Cycle number of directory

ARG 64

File server (reply port):

0	Receive block (shown on summary page)
4	
5	
	Creation date in standard format Byte 5 - Day Byte 6 (Top 4 bits) - Year since 1981 (Bottom 4 bits) - Month Byte 7 - Hour Byte 8 - Minutes Byte 9 - Seconds
10	
	Last update in standard format Byte 10 - Day Byte 11 (Top 4 bits) - Month (Bottom 4 bits) - Year since 1981 Byte 12 - Hour Byte 13 - Minutes Byte 14 - Seconds
15	

## Set Object Attributes

Function Code=19

### General description

This call is used to set the attributes for a file or directory. This is used by OSFILE. Note ARG=64 is only supported on SJ Research File Servers.

### On entry,

0	Transmit block (shown on summary page)
7	
	ARG
	1 = set load/execute/access
	2 = set load address
	3 = set execute address
	4 = set access
	5 = set creation date (only 2 bytes)
	64 = set update & creation date and time (10 bytes)
8	Parameters to set (depend on byte 7)
n	
v	File name terminated by CR

### On exit,

0	Receive block (shown on summary page)
4	

## Delete Object

Function Code=20

### General description

This call is used to delete an object and is used by OSFILE. It is not possible to use wildcards with this call.

### On entry,

0	Transmit block (shown on summary page)
7	
n	Object name terminated by CR

### On exit,

0	Receive block (shown on summary page)
4	
8	Load address
12	Execute address
15	Size information

## Read User Environment

### General description

This call returns your current position on the file server.

#### On entry,

0	Transmit block (shown on summary page)
7	

#### On exit,

A,X,Y

0	Receive block (shown on summary page)
4	Length of disc name (16)
5	Name of currently selected disc padded with spaces
21	Name of CSD padded with spaces
31	Name of LIB padded with spaces
41	

Function Code=21

## Set User Boot Option

### General description

This call writes your boot option to the password file. This call is used by \*OPT 4.

#### On entry,

0	Transmit block (shown on summary page)
7	
8	boot option

#### On exit,

0	Receive block (shown on summary page)
4	

Function Code=22

## User log-off

Function Code=23

### General description

This call is equivalent to a \*BYE packet to the fileserver, except that it closes open print files as well. The File Server, on receiving the command, removes the station number from its internal table, closes all file handles and context handles open for that station number.

### On entry,

0	Transmit block (shown on summary page)
7	

### On exit,

0	Receive block (shown on summary page)
4	

## Read User Information

Function Code=24

### General description

This call returns the first station number in its user table that matches the user identifier provided. Note that on an SJ Research file server the station number returned will always be the machine which the user most recently used, this is not the case with the Acorn file server.

### On entry,

0	Transmit block (shown on summary page)
7	
n	
	user name terminated by CR

### On exit,

0	Receive block (shown on summary page)
4	
	0 = unprivileged non-zero = privileged
5	station number (low byte first)
7	

## Read FS Version Number

Function Code=25

### General description

This call returns a version number string. It is not necessary to be logged onto the file server to do this call. This function is used in the SJ Research library utility FSLIST.

#### On entry,

0	Transmit block (shown on summary page)
7	

#### On exit,

0	Receive block (shown on summary page)
4	
n	Character string describing version number

## Read FS Free Space

Function Code=26

### General description

This call returns the size and current amount of free space on a file server disc. This call is used by the library utility \*FREE.

#### On entry,

0	Transmit block (shown on summary page)
7	
n	Disc name (terminated by a CR)

#### On exit,

0	Receive block (shown on summary page)
4	
7	Free space on disc (low byte first)
10	disc size (low byte first)

## Create Directory

### General description

This call creates a directory specifying the number of blocks, for the file server, to allocate to that directory. One block is 256 bytes.

#### On entry,

0	Transmit block (shown on summary page)
7	
8	Number of blocks to allocate
n	Directory name (Terminated by a <CR>)

#### On exit,

0	
4	Receive block (shown on summary page)

Function code=27

## Set Real time clock

### General description

This System privileged command allows the Date and Time to be set, and is used in the Library utility program 'settime'. This command is normally privileged on SJ Research File Servers, but this may be changed, by using function code 65.

#### On entry,

0	Transmit block (shown on summary page)
7	
9	Date in standard format Byte 7 - Day Byte 8 (Top 4 bits) - Year since 1981 (Bottom 4 bits) - Month
12	Time in standard format Byte 9 - Hour Byte 10 - Minutes Byte 11 - Seconds

#### On exit,

0	
4	Receive block (shown on summary page)

Function code=28

## Create file

Function code=29

### General description

This follows the same protocol as Command code 1, 'SAVE'. However the data transfer phase is omitted. The result is that a requested amount of space is reserved for a file, the data therein being undefined. This call is not supported on early versions of File Server software.

## Read User Free Space

Function code=30

### General description

This call is the Acom file server call to read a users free space, the call is the equivalent of the SJ Research accounting system (detailed in function code 64).

### On entry,

Client (command port),

0	Transmit block (shown on summary page)
7	
	User identifier for free space interrogation (terminated by <CR>) A null user identifier means return information about client
n	

### On exit,

0	Receive block (shown on summary page)
4	
8	Available space for user identifier (in bytes)

## Set User free space

Function code=31

### General description

The function sets the amount of space available for a user identifier. The function is only legal for system privileged users. The user identifier is that of the client whose space allocation is to be ammended. This call is only implemented on Acom File Servers.

### On entry,

0	Transmit block (shown on summary page)
7	
	New amount of free space
11	
n	User identifier for free space interrogation (terminated by <CR>)

### On exit,

0	
4	Receive block (shown on summary page)

## Read client User Id.

Function code=32

### General description

Reads the username which you used to logon to the File Server with. This call is not supported on early version of File Server software.

### On entry,

0	
7	Transmit block (shown on summary page)

### On exit,

0	Receive block (shown on summary page)
4	
n	User identifier terminated by a CR

## Read Account information

### General description

This call reads the amount of space left in an account.

#### On entry,

0	Transmit block (shown on summary page)
7	ARG=0
8	First account to try
10	Maximum number of accounts to send information on
12	Disc number
13	

#### On exit,

0	Receive block (shown on summary page)
4	Next account to try
6	Number of accounts returned
8	1st account number
10	1st account space
12	2nd account number
14	2nd account space
16	.
	.
	.

N.B. This call is only supported on the SJ Research file server.

Function code=64

## Read/write system info.

Function code=65

### General description

This call provides an interface to read and write the printer information and change the privilege needed to write the file server's time. All of these read calls are unprivileged commands. All the write operations are privileged.

#### Reset print server information :

This call is used to reset the printer information and **must** be issued for any of the other change printer information calls to take effect.

#### On entry,

0	Transmit block (shown on summary page)
7	ARG=0
8	

#### On exit,

0	Receive block (shown on summary page)
4	

#### Read current state of printer :

This call returns the detailed information about a logical printer.

0	Transmit block (shown on summary page)
7	ARG=1
8	Printer number (1 - 8)
9	

### On exit,

0	Receive block (shown on summary page)
4	Name of printer (padded with spaces)
10	Bit 3 - Spool to disc Bit 2 - Account ownership required Bit 1 - Anonymous users allowed Bit 0 - Printing enabled
11	Account number (Only relevant if bit 2 of previous byte is set)
13	Banner file name (terminated by a CR if less than 23 characters)
n	

### Write current state of printer

This call writes the detailed information about a printer, system privilege is required to do this.

### On entry,

0	Transmit block (shown on summary page)
7	ARG=2
8	Printer number (1 - 8)
9	Name of printer (padded with spaces)
15	Bit 3 - Disable spooling to disc Bit 2 - Account ownership required Bit 1 - Anonymous users allowed Bit 0 - Printing enabled
16	Account number (Only relevant if bit 2 of previous byte is set)
18	Banner file name (terminated by a CR if less than 23 characters)
n	

### On exit,

0	Receive block (shown on summary page)
4	

### Read the AUTO printer priority

This call reads the order in which printers are selected for users who have not requested a particular printer.

### On entry,

0	Transmit block (shown on summary page)
7	ARG=3
8	

### On exit,

0	Receive block (shown on summary page)
4	Number of printer entries available (current implementation=2)
5	1st choice of printer
6	2nd choice of printer
7	
n	

### Write the AUTO printer priority

This call allows a privileged user to write the order in which printer are selected for users who have requested the AUTO printer.

### On entry,

0	Transmit block (shown on summary page)
7	ARG=4
8	Default printer 1
9	Default printer 2
10	
(n+8)	Default printer n
(n+9)	

### On exit,

0	Receive block (shown on summary page)
4	

### Read system message channel

This call returns the physical printer that all system messages are sent to. Note that the printer is a physical printer, so the parameter should be either 1 (parallel) or 2 (serial).

### On entry,

0	Transmit block (shown on summary page)
7	
8	ARG=5

### On exit,

0	Receive block (shown on summary page)
4	
5	Current system message printer

### Write system message channel

This call allows a privileged user to set the physical printer that system messages come out of.

### On entry,

0	Transmit block (shown on summary page)
7	
8	ARG=6
9	New system message printer

### On exit,

0	Receive block (shown on summary page)
4	

### Read message level

This call reads the current level of system messages. The value returned is in the range of 0 to 255. The amount of output is the level of output selected plus all the levels below that level. Therefore, in the list of levels shown to set the message level to 7 would make the file server print all logons and logoffs as well as errors.

### Message level

### Description

0	Off
5	Logon/logoff
7	Errors (i.e. 'Wrong password', 'bad name' etc.)
10	Maximum users and all star commands
11	Load/save
15	*cat and opens
128	Aborted loads
130	Function codes
150	Network errors
170	Map building names
200	Disc read/write
250	All successful network transactions to and from the fileserver
255	All Activity to the JPROC processor

### On entry,

0	Transmit block (shown on summary page)
7	
8	ARG=7

### On exit,

0	Receive block (shown on summary page)
4	
5	Current message level

### Set message level

This call sets the message level, as described above. It should never be necessary to set the message level to greater than 127 and that setting the message level to a value greater than 150 produces excessive output and will probably reduce the performance of the file server.

### On entry,

0	Transmit block (shown on summary page)
7	
8	ARG=8
9	New message level

### On exit,

0	Receive block (shown on summary page)
4	

## Read the default printer

This call returns the default printer. This printer will be selected if a user starts to print without having selected a particular printer.

The values for the default printer are :-

- 0 Automatic search through the list default printer priority list (set by Fn=65 ARG=4)
- 1 Logical printer 1
- 2 Logical printer 2
- ...
- ...
- 8 Logical printer 8

255 Hold the job output in the %PRINTQ directory.

### On entry,

0	Transmit block (shown on summary page)
7	
8	ARG=9

### On exit,

0	Receive block (shown on summary page)
4	
5	Current default printer

## Set the default printer

This privileged call sets the default printer, see ARG=9 for more information about valid printer numbers.

### On entry,

0	Transmit block (shown on summary page)
7	
8	ARG=10
9	New default printer setting

### On exit,

0	Receive block (shown on summary page)
4	

## Read the privilege required to change time

This call reads the privilege required to change the file servers time. The SJ Research file server normally insists on system privilege to be able to change the time. However, if it is desired to change the time frequently this can be disabled.

### On entry,

0	Transmit block (shown on summary page)
7	
8	ARG=11

### On exit,

0	Receive block (shown on summary page)
4	
5	0 - Privilege required Non zero - Privilege not required

## Set the privilege required to change time

This call sets the privilege required to change the file servers time.

### On entry,

0	Transmit block (shown on summary page)
7	
8	ARG=12
9	0 - Privilege required Non zero - Privilege not required

### On exit,

0	Receive block (shown on summary page)
4	

N.B. These calls are only supported on the SJ Research file server.

## 10.22 Password File Format

The password file, and the user information stored by the fileserver, is stored in the file %PASSWORDS. The following information is included for users who wish to write their own password handling programs, such as implementing the Acom File Server commands 'NEWUSER', 'REMUSER' and 'SPRIV'. It follows the following format :-

0	Entry number of 1st user entry with the first character less than 'A'
2	Entry number of 1st user entry with the first character as 'A' or 'B'
4	Entry number of 1st user entry with the first character as 'C' or 'D'
6	.
28	Entry number of 1st user entry with the first character greater than 'Z'
30	Entry number of the default user
32	Not used
64	1st User entry
128	2nd User entry
192	3rd User entry
256	.
n	.
(n+64)	Terminating User entry (Filled with &FF)
z	URD names and Libraries (pointed to by the user entry) Each a maximum of 80 characters terminated by a <CR>

Each user entry (occurring in the password file every 64 bytes) has the following format :-

0	User Identifier (Terminated by a <CR> if less than 10 characters)
9	Password (Terminated by a <CR> if less than 10 characters)
19	Boot option
20	Flag bit 0 = Password unlocked bit 1 = System privileged bit 2 = No short SAVES bit 3 = Permanent *ENABLE bit 4 = No Library bit 5 = Run only user
21	Offset from start of file to user's Root Directory Set to 0 if URD is normal
24	Offset from start of file to user's Lib. Directory Set to 0 if LIB is normal
27	Personal account number
29	Reserved
32	Bit map of account ownership bit 0, byte 0=Account 0 bit 7, byte &1F=Account 255
64	

This information is not available on Acom File Servers.

## 10.22 Application Notes

Topics covered:

OSARGS  
Printing  
Econet Workspace  
Use of OS workspace (or, How to make games work)  
The program NETMON  
Basic Guide to Econet Line Protocols

### Glossary of terms and abbreviations used in this section:

AUG           Advanced User Guide for the BBC Microcomputer - Bray, Dickens & Holmes.  
ESUG           The Econet System User Guide (The greyish book).  
EAUG           Econet Advanced User Guide (The black book with sky and clouds on it).  
UG            BBC Microcomputer User Guide - beware, details of OSFIND etc are WRONG.  
OSHW           Operating System High Water Mark -- A system variable, to which BASIC sets the value of PAGE.

The value of OSHWM depends on how many filing system ROMs are present in a BBC machine. (Languages do not take up any room, except when they are active). Here is a list of some common configurations:

ROMs fitted value of  
  OSHW

No ROMs &0E00  
Disc &1900  
Econet &1200  
Disc & Net &1B00  
Teletext &2200  
Teletext & Disc &2400  
Teletext & Econet &2400  
Teletext, Disc & Net &2600

### File names

Although individual components of an Econet file name may not be longer than 10 characters, it is quite possible to have a compound file name (eg \$JOHN.BBCprogs.....2ndrev.OLD.developmnt.Addition) of almost infinite length. A compromise length of 80 characters should be used as a minimum. Even a DFS file name can be up to 12 characters long (:2.R.LONGEST), requiring 13 bytes of storage including the CHR\$(13) on the end. Programs which disallow filenames over 7 characters long are highly unsatisfactory.

### Econet workspace

When writing \* command programs it is desirable to allow them to run in areas of memory which do not corrupt BASIC programs and other valuable user- or system data. The most obvious way of doing this with Econet-specific commands is to use the Econet workspace (Public area) which comprises pages &E and &F. When writing Filing-system independant commands use the area mentioned in note b) below.

As a general guide the following areas are safe:

E10..E1D	OK
E1E..E22	Corrupted during loading
E23..E2F	OK
E30	? DNFS stores file name
E31..EFF	OK
F00..F03	used by * commands, OSFILE etc.
F03..F04	must be zero
F05..F08	Corrupted during loading
F09..F0C	must contain the 32-bit execute address
F0D..FDC	OK

Notes:

a) F00 is used as a buffer for all FS-related commands eg \*I AM, \*<filename>, OSFIND etc.  
b) It seems now to be an accepted convention that if you can't get \* commands to work in the Econet workspace, then pages 9 and 10 (ie &900..&AFF) should be used.

### Zero-page workspace

If you are writing your own \* commands, and need some zero-page workspace, it is important to use the correct area. There is an area specifically reserved for such 'Operating System' commands, and this is at locations &A8..&AF. See AUG p. 268.

NB it is not good enough to use the 'spare' BASIC workspace &70..&8F, as there is no reason why a user may not run any \* command from within any language, not necessarily BASIC.

### Use of OS workspace

The area of memory below PAGE (or to be more precise OSHWM) is reserved for use by the OS and any filing systems that may be in your machine. Interfering with this memory can have unexpected and disastrous effects. However, many games and other (illegal, in the bad programming sense of the word) programs use RAM below PAGE, sometimes as low as &400 (an admittedly exceptional case).

On the Econet this is particularly nasty as the NFS uses NMI which will interrupt ALL machines whenever any net traffic occurs. This results in machines crashing when plugged into the network. In fact it is not only the NMI workspace which is corrupted during an interrupt but also the NFS Public workspace (pages &E and &F) and the NFS Private workspace, which effectively means any part of RAM below OSHWM.

The solution is for the user himself to 'claim' the NMI workspace, which has the effect of disabling the Econet altogether. It is best documented in the AUG pp. 320.

To claim NMI workspace use OSBYTE &8F, viz:

A%=&8F (sideways ROM call)  
X%=&C (the 'claim' call)  
Y%=&FF (as it instructs you to do)  
CALL &FFF4

or \*FX143,12,255

Of course the main disadvantage of disabling the Econet is that you cannot use any of its facilities while the program in question is running. If you are writing your own software, disabling the Econet should only be considered as a last-ditch option. Educational users will be particularly annoyed as the majority have networks and may wish to use network related utilities with such software.

## 10.23 The Four-way Handshake

Data is transmitted across the network using a four way handshake. This transfer protocol is done automatically by the NFS ROM and the user need not be aware of its existence.

The first packet, sent by the sender machine, contains the destination station number, the source station number, the port on which the data is being sent, and the type of operation. This packet is called the 'scout packet'. The second packet, which is called the 'acknowledge packet', is sent by the destination station, if the following conditions are met :-

- 1) The machine is on the network.
- 2) The machine has a receive block open for that port.
- 3) The machine is not protected against the operation.

The third packet is the sender machine sending the data. The fourth packet is the destination machine's acknowledgement and means that the data has been correctly received.

All operations use this four way handshake with the exception of peek, machine type peek, halt, continue and broadcast. The four-way handshake can be seen using the utility 'NETMON', which is documented below.

## Description of the Program NETMON

NETMON provides a means of continuously monitoring the network at a very low level. It displays the data bytes as they are sent down the network and also some status information, particularly useful in the debugging of network software, and networks in general.

\*NETMON loads some very special code which runs the network hardware directly. This effectively removes it from the network, as far as other machines and the program \*STATIONS is concerned. It is wise after running NETMON to power-off before attempting to use it for normal programming purposes.

The program prints:

```
ECONET MONITOR xxx  
1E
```

where xxx is the station number of the monitor machine. Monitor output can at any time be stopped by pressing the space bar (ctrl-shift functions as normal).

Data bytes as they are sent on the network are printed in hex. There are various statuses that are printed and these are:

<space>

Address present.

Indicates that the next byte is an address byte, which is always the first of a packet. Packets are therefore always separated by spaces.

! <newline>

Idle detected.

Occurs between any two Econet messages. Stations wishing to transmit must always wait for an idle condition on the line (a sequence of 15 one's) before enabling their line drivers. A long string of i's is caused by network hardware problems, usually either poor wiring, or one or more blown SN75159(BBC)/26LS30(Master) line driver ICs.

- v        **Frame valid.**  
Occurs just before the last byte of a packet, indicating that the CRC was valid.
- e        **CRC error.**  
This can be caused by two stations transmitting simultaneously, or by noise getting into the network, or by an intermittent network connection, or a bad econet lead anywhere on the network, or....
- o        **Data overrun error.**  
At clock speeds above 160KHz or thereabouts the monitor cannot keep up with the data rate. However, BBC machines are quite capable of running up to about 235Kbaud (2nd processors just under 200Kbaud reliably). Unless loss of data bytes, by the monitor program, is a nuisance, this should be of little concern.
- b        **Abort.** Normally an error, but at high clock speeds these can occur on a correctly functioning network (the b occurs in place of the v). One can also get packets like:  
  
C800010080b i  
99  
  
where it would appear that an idle has occurred in the middle of a packet! This is due to the receive FIFO register in the SDLC chip, which buffers the data bytes but not the statuses. In fact in real time the idle occurred after the data bytes.
- d        **Clock missing.**  
Lots of d's indicates that the clock (to the monitor station at least) is intermittent. Suspect Econet lead, or clock connection in the network, or clock itself.

## A Basic Guide to Econet Protocols.

Hex numbers are preceded by &, all other numbers are in decimal.

A standard Econet interchange might look like this:

```
FE00120080v99 1200FEv00 FE001200900301010203000Bv0D 1200FEv00 i
```

which is one of the messages sent when doing a \*CAT.

Each of the groups of numbers separated by a space is called a *Packet*, and a group of packets constituting one of the legal Econet transfer protocols is called a *Message*. The packet is the basic unit on the Econet. Packets consist of:

- One or more flags (not displayed on monitor)
- Any number of data bytes
- 2-byte CRC        (only correctness or incorrectness displayed on monitor)
- One flag        (not displayed on monitor)

Data bytes within the packet follow with no gaps and there are special encoding techniques which ensure that the flag pattern does not occur within a packet. The packet structure is known as SDLC, and it is constructed and decoded in hardware by a chip in the Econet circuit. On the BBC machine an MC68B54 is used (IC 89) and on Z80-based machines a Z80 SIO.

You will notice that each packet has the same 4 bytes on the front but the order changed. All Econet packets have a 4-byte header describing where the message is going, and who sent it. Both these 'addresses' are 2-byte quantities, the first byte being a station number and the second a gateway number. Normally the gateway number is zero, unless you have a *Bridge* on your network in which case this may be non-zero. A zero gateway number addresses your 'local' network.

The first 'address' is the *destination* address. This is on the front of a packet so that any machine (all machines listen all the time) can tell whether that packet is directed at it or at another machine. The second address is the *source* address, ie the network address of the machine which transmitted the packet. In the first packet FE00 was the destination address and 1200 was the source address.

You should see now that, in the example above, 2 packets were going from station &12 to station &FE, and two from station &FE to station &12.

Let us now examine the message in more detail, packet by packet.

The first packet is a *scout packet*, sent to station &FE (=254, so probably the FS) from station &12 (a client). In addition to the packet header, there are two bytes; the first is a *control* byte, the second the *port* number identifying the rest of the message. (NB the word port here is nothing to do with hardware ports, although it has much the same function as an identifier.) The control byte isn't very important as regards the FS interface: it can be used for sequencing and is used extensively for printing.

The port number is much more interesting. Stations which are set up to receive messages can selectively allow messages from <all stations or one particular station> and on <all ports or a specified port>. The port number identifies the data to the receiving station as 'this is the data I wish to SAVE' or 'here are some bytes to be printed' or 'this is a print status enquiry': &99 has identified the message as an *FS command*.

The second packet is an acknowledgement. The first packet was sent from the client to the FS. The acknowledgement packet goes the other way (notice the header bytes are the other way around) and it is a

packet telling station &12 that station &FE has recognised the scout package and is prepared to receive some data.

The third packet is a data packet. After the header there follow data bytes. The protocol does not specify in advance how many bytes are going to be sent, and it is up to the transmitter (station &12) to decide how many he is going to send. If too many are sent an error will be reported at both ends. Here are the data bytes again (less the packet header):

```
90 03 01 01 02      03 00 0B 0D
```

To interpret these bytes we first have to remember the port number in the scout packet, which was &99, the FS command port. This defines the first 5 bytes of data to be a *Standard Tx Header* (See page 63 of EAUG, page 99 of ESUG), so that the reply port is &90, Function code is 3 (an 'Examine' call, used by \*CAT) and context handles are 01, 01 and 02.

The rest of the data bytes are parameters to the examine call (p 65 of EAUG, p.105 of ESUG), which gives ARG=3, and requests &0B entries starting from entry 0 in directory "" (ie the CSD). An ARG of 3 tells the FS that the data returned should be file title + access, in ASCII.

The fourth packet is also an acknowledgement. It tells stations &12 that stations &FE has received the data correctly, and that not too much data was sent.

We have looked at a successful Econet transfer. It is also wise to know about transfers that didn't work; the most likely of which is when the error reported is *Not listening*. This can happen in various ways; either the distant machine is not present, or switched off, or it has not been set up for receive or it may have crashed. In either case the monitor output will look like this:

```
FE00120080v99 i
FE00120080v99 i
FE00120080v99 i
FE00120080v99 i
FE00120080v99 i
FE00120080v99 i
```

ie station &12 repeatedly sending scout packets to station &FE but getting no acknowledgement (It sends a few hundred such packets before reporting an error).

This sort of monitor output may also occur under perfectly normal circumstances: if an FS is busy with another client and/or there is disc activity going on it will not be set up to receive packets on its command port, and so will not send any acknowledge packets.

A less likely form of error may look like this:

```
FE00120082vD1 1200FEv00 FE001200000D0A03v31 i
```

(again many times). Station &FE has acknowledged the scout packet from station &12, but has not acknowledged the data packet. Almost certainly too many bytes were sent (the BBC machine will report *Net error*)



File Server is running. The BBC Microcomputer expects the File Server to be on Station Number 0.254 unless you tell it otherwise (some games and similar software can accidentally do that). If you wish to use a different File Server, or to change back to File Server 0.254, you will have to tell the computer explicitly. See section 3.4 for details, also Section 9.4 for further information on network debugging.

No clock

Error A3 (Decimal 163)

There are three possible causes of this message. First, and most likely, the computer is not plugged into the network. Check that the Econet plug in the back of the computer is plugged in firmly, and that the other end of the lead is plugged into the wall socket or cable adaptor that connects to the network.

Second, there must be (only) one clock in the network. It is possible that it has been unplugged from the mains or from the network, or that there is a fault in the clock unit. Check that it is plugged in correctly. If this is not the problem, connect the clock unit to only one BBC Microcomputer. Press N and BREAK as before on this machine: if the "No Clock" message still appears, suspect the clock unit.

Third, there is a fault on the network line itself. This is unlikely to happen if the network has functioned before, but could happen as a result of mechanical damage to the cable, or a fault in one of the machines attached to the network, or in one of the network terminators. Remove these items one by one until there is no "No Clock" message when you press N and BREAK. (Note that the message will not change on its own: you have to press N and BREAK).

Station nnn.xxx not present Error A4 (Decimal 164)

An error message from an Acorn computer with the advanced network filing system ROM indicating that the last file server command was sent to a file server which either did not exist or was not on line.

No reply

Error A5 (Decimal 165)

The "No reply" message will occur if a filing operation fails in the middle of the operation. This error will occur if the file server runs out of space in the %PRINTQ directory whilst print spooling.

## A.2 File Server Errors reported as Error A8 (Decimal 168)

Password file changed

FS Error 03 (Decimal 3)

A user has attempted to set a password (using \*PASS) or boot option (using \*OPT 4), and the password file has been edited since the user logged on.

Bad number

FS Error 04 (Decimal 4)

A number has been incorrectly specified. The number may be in the wrong base, larger than the maximum value allowed, or simply not supplied at all.

Key locked

FS Error 05 (Decimal 5)

Only occurs if a system privileged user attempts to use a system privileged command when the front panel key-switch is not in the "SYST" position. Only appropriate to Hard Disc and Modular Disc File Servers.

Too short

FS Error 06 (Decimal 6)

If the system manager has set the appropriate flag in their password file entries, users are prevented from SAVEing files shorter than 16 bytes. This is in order to guard against the possibility of pressing <Break> before saving a program, and then saving a null program over the previous copy, for example. If it is necessary to create a short file, this may be done by means of OPENOUT and BPUT (see Section 3.4).

It is possible to save short files by typing **\*ENABLE** (conversely typing **\*DISABLE** will prevent files under 16 bytes being saved). When you logoff this will be reset to the one in your own passport file.

Circular RENAME

FS Error 07 (Decimal 7)

Whilst directories may be renamed into other directories, it is not possible to rename a directory into itself or one of its sub-directories as this would create a circular structure of directories.

Printer busy with station xxxx FS Error 09 (Decimal 9)

Generated by **\*PRINTOUT**, if a non-spooling logical printer is not free.

Not authorised to use printer FS Error 0A (Decimal 10)

Generated by **\*PRINTOUT**, if the system manager has restricted the use of this logical printer to holders of a certain account (See Section 4.3, **EDITPRINT** program)

File too big

FS Error 35 (Decimal 53)

Files may not exceed 8 Megabytes in length.

Illegal attribute

FS Error 46 (Decimal 70)

An attempt has been made to set an illegal combination of attribute bits: setting read or write access to a directory, for example.

Bad ARG to examine

FS Error 4F (Decimal 79)

The argument to one of the examine (catalogue) calls is out of range.

Bad ARG to read arguments

FS Error 6D (Decimal 109)

The argument to the call to read random access information is out of range.

Not supported

FS Error 85 (Decimal 133)

The File Server function code is not one of the recognised values.

Bad time

FS Error 90 (Decimal 144)

The set time function has been called with a time or date that is invalid (eg. 32nd August).

File Server Offline

FS Error A2 (Decimal 162)

The File Server is offline and waiting to be booted up, or taking a backup, in Utility Mode.

### A.3 The following File Server Errors are reported with their correct Error numbers

Note that some of the errors have the same error number, despite having different text messages. This is to assist compatibility with Acorn and other systems -- the error number has been made the same as an existing one, when the action required (by a program, for example) to recover from the error would be similar. For example, **Disc full** and **Account xxx bankrupt** would typically require space to be made by deleting something, hence these both have error number C6.

Error A8 (Decimal 168)

"Catch-all" error number, see A.2 above.

Not logged on

Error AE (Decimal 174)

The specified user is not logged on to this File Server. This error arises from utilities such as \*NOTIFY FRED.

Renaming across discs

Error B0 (Decimal 176)

It is not possible to transfer a file from one disc to another by use of \*RENAME. The standard method for renaming a file between discs is to copy the file (using COPIER) and then to delete the old copy.

Directory full

Error B3 (Decimal 179)

A directory may not contain more than 255 entries.

Directory not empty

Error B4 (Decimal 180)

It is not possible to delete a directory unless it is empty (ie. contains no files or sub-directories).

xxxx is not a file

Error B5 (Decimal 181)

It is not possible to LOAD a directory or to open it for output. While it is possible to open a directory for input, any attempt to use BGET will cause this error (see \*ACCESS in Section 3.4). The error may also be caused by an attempt to create a file of the same name as an existing directory.

Too many users

Error B8 (Decimal 184)

The File Server has a maximum number of users (typically 60). The File Server will not log stations off automatically, so it would be possible to log on, then at some stage change the station number. The File Server user list would then contain a user logged on at a non-existent station. The solution is to either stop and re-start the File Server, or use \*LOGOFF (see section 4.3) to remove stations from the list.

Bad password

Error B9 (Decimal 185)

Passwords have the same restrictions as file names concerning illegal characters. In addition, no wildcards are allowed.

Insufficient privilege

Error BA (Decimal 186)

This error occurs when a user without system privilege attempts to perform some privileged operation, such as editing the password file. This error can also be caused by a user attempting to change his password or boot option, if this has been prohibited by the system manager.

Wrong password

Error BB (Decimal 187)

The password specified in the \*I AM or \*PASS command does not match that stored in the password file.

User not known

Error BC (Decimal 188)

If a user attempts to log on, and his user identifier does not appear in any password file in the system, and no default user has been set by the system manager, then this error will be produced. It is most commonly a result of mis-spelling the user identifier.

Insufficient access

Error BD (Decimal 189)

The user does not have sufficient access to the file or directory in question. If the main or auxiliary account numbers occur in the list of accounts associated with the user he has 'Owner' access, otherwise he has 'Public' access. Random access reading or writing of files is controlled by the W and R access attributes: those before the stroke apply to owners and those after the stroke to public users (see \*ACCESS in Section 3.2).

To create a new file, the user must have owner access to the directory in which it is to be created.

To SAVE a file over one of the same name, owner access is required to the file and it must not be locked (L attribute), but the W and R attributes are not checked.

Users may check the access attributes and account numbers of files or directories by use of \*INFO (the accounts are the last two items displayed), and may check their own list of accounts by use of the \*STATEMENT utility.

Not enabled

Error BD (Decimal 189)

The system manager can set an option to require a user to type \*ENABLE before using \*DELETE with a wild card. This is to prevent inexperienced users from deleting files accidentally. Once \*ENABLE has been typed, its effect remains until \*DISABLE is used, or until the user logs on again.

xxxx is not a directory

Error BE (Decimal 190)

The name of a file has been specified in a context where a directory is required - for a catalogue or as the directory to search for a file (e.g. in a \*DIR command).

Who are You?

Error BF (Decimal 191)

The user is not logged on to the File Server, and should use the \*I AM command to log on.

Too many files open

Error C0 (Decimal 192)

There are 8 channels available, of which two are permanently allocated to the Library and User Root Directory. In addition, the currently selected directory may require one, and another is required during the execution of the \*DIR command. The remainder may be used for random access files.

File not open for update

Error C1 (Decimal 193)

The random access file in question has been opened for reading only (e.g. using OPENIN from BASIC), followed by an attempt to write to the file.

Already opened by xxxx

Error C2 (Decimal 194)

The file or directory has already been opened by the specified user (and possibly by others). It is not possible to delete files or directories which are open, nor to write to files. It is possible to read files which are already open, so long as they have been opened for input only. Note that each user is considered to have

opened his User Root Directory, Library, and Currently Selected Directory.

Locked

Error C3 (Decimal 195)

An attempt has been made to DELETE, SAVE over, OPENOUT over or RENAME a file or directory which is locked (access L is set). Use \*ACCESS <name> -L to unlock it.

Already exists

Error C4 (Decimal 196)

An attempt to RENAME a file when a file already exists with the new name is not allowed.

Disc full

Error C6 (Decimal 198)

There is insufficient free space on the disc. Note that there may be unexpected overheads if saving a file requires a directory to be extended, and that a file longer than 16K incurs an overhead of 1K for each 512K of extent. It is possible to create files of large extent where the data blocks do not all exist - unwritten blocks will read back as zeros, and disc space must be found when they are first written. All disc space allocation is in units of 1K.

Account xxxx bankrupt

Error C6 (Decimal 198)

The account to be charged for the current operation does not hold sufficient credit. Accounts are charged in the same way as disc space is allocated: see notes under 'Disc Full' concerning overheads. When a file is created, it is charged to the account of the directory in which it exists. If \*ACCOUNT is used subsequently to change the account, the disc space taken by the file will be charged to the new account and re-credited to the old; the space taken by the directory entry is still charged to the account of the directory.

Drive Error

Error C7 (Decimal 199)

A media error occurred while trying to read the disc or tape. Disc errors have serious consequences for the integrity of the system and should be reported to the system manager immediately, as they may indicate an impending problem with the disc or disc drive.

Disc changed

Error C8 (Decimal 200)

The disc has been changed, on a Floppy Disc (or RM380Z) File Server. Log on again with \*I AM. (This error is never generated by SJ Research File Servers, since "Who are you?" is more helpful.)

Drive read only

Error C9 (Decimal 207)

There are four possible reasons for this error. On a floppy disc, this error can be caused by a write protect tab on the disc to which the user is attempting to write. (If the system manager is going to remove the tab, he *must* go through the normal disc changing procedure). On any type of disc this error may indicate that the File Server has not finished scanning the directory when booting the disc as a consequence of the directory scan failing as a result of a corrupt directory structure (this will be accompanied by the message Bad backpointer on drive n or Wrong number of files in DIR). The fourth possible reason is that you are trying to write to the tape whilst using %TAPE.

Bad name

Error CC (Decimal 204)

The filename used is illegal. Filenames may not contain the characters \$ % . ^ & or : except where they are used for their special meanings - see Section 3.1.2. Characters above 126, control characters and spaces are not allowed at all.

Bad wildcard

Error CC (Decimal 204)

A wildcard character has been used where this is not allowed. Wildcards are not allowed in the filename used to create a file (although they may always appear in the directory name), or in passwords. This error will also occur if the number and types of wildcards do not match in the two filenames in a \*RENAME command. You will also get this error if you try a wildcard delete on a Master series microcomputer; to get round this type **\*\DELETE** .

Bad attribute

Error CF (Decimal 207)

One of the attributes specified in a \*ACCESS command is not one of the permitted attributes M P D L / W or R.

xxx not found

Error D6 (Decimal 214)

The specified file, disc or directory could not be found. Note that this error will be caused if you try to gain access to someone else's file which has been set to Private in the \*ACCESS command.

Channel

Error DE (Decimal 222)

The channel number specified does not refer to a file which is open. This error can be caused by closing a file channel too soon, but the most common reason is that the 'context handles' have been lost in the client machine. These handles are required for almost all File Server operations, and can be lost by switching off the computer, or using software which overwrites memory illegally (many games do this!), or due to various bugs in the Acorn NFS ROM. The solution is to log on again with \*I AM xxx.

EOF

Error DF (Decimal 223)

An attempt has been made to read data beyond the end of a file.

Bad string

Error FD (Decimal 253)

The text string supplied is illegal, due to mis-matched quotes or an incomplete | (bar) sequence.



# Appendix B: System errors

---

## Contents:

B.1	General Information
B.1.1	Examples
B.2	File Server Messages and Internal Errors
B.2.1	Non-fatal Printed Errors
B.2.2	Errors denoted by System Error LED
B.2.3	Flashing System Error LED
B.3	Winchester Disc Errors
B.3.1	RODIME RO752 Errors
B.3.2	Adaptec ACB 4000A Errors
B.4	Tape Errors
B.5	SCSI Sense Key Definitions
B.6	Floppy Disc Errors

## B.1 General Information

Errors generated by the file server fall into 3 classes:

- a) 'Warnings' which do not inhibit further operation of the File Server
- b) Fatal errors which put the File Server 'Offline' (usually accompanied by the System Error LED)
- c) Errors caused by failure to read or write one of the discs in the Fileserver.

When On line, error messages are sent to one of the printer ports on the MDF5. Which one it is is set up by **EDITPRINT**. It is wise therefore to always keep a printer connected to the system even if no print output is anticipated. In Utility Mode, error messages are only sent to the screen of the BBC microcomputer running \*FAST.

There is another error condition on the MDF5, which can occur at power-up and is denoted by a flashing System Error LED.

### IMPORTANT NOTICE:

- 1) It is wise to keep a permanent copy of any error messages printed by the File Server. This is very important in the case of errors other than disc errors, and you should try and record all the information printed. It may be valuable information at a later date when sorting out a particular problem.
- 2) Try to ascertain who was using the File Server at the time, and what they were doing.
- 3) Ring SJ Research as soon as possible (on 0223 69927).

### B.1.1 Disc & Tape Error Examples

The general form is:

```
Wini|Floppy|Tape [read|write] error [on drive <letter>], block <hhhh>
```

For example:

```
Drive 01 read error 81 at block 0002
Floppy read error 08 on drive B, block 02F5
Wini read error 91 on drive E, block 3001
Floppy write error 10 on drive A, block 0001 ** SERIOUS ERROR **
```

Write errors are considered 'serious' because they indicate that a disc crash will probably occur if the disc is used for very much longer. The best course of action is to recover any vital files and save them somewhere else, and then avoid using that disc again.

```
Wini error 92 on drive F, addr=00A69B
```

This example would only be printed when running the MDFS Utility Mode. In this case the 00A69B refers to a 'SCSI logical block number' and not a File Server block number. The block number is printed in this format because it is more useful when *reallocating* bad sectors on a winchester (see section 7.3.5). You can recognise this format because there will always be 6 digits as opposed to 4.

Note the drive *number* which indicates a logical drive (the first logical drive is drive 00). If you had a single winchester and some floppy drives connected, **Drive 01** would refer to the first floppy drive. This error probably indicates that someone had removed the disc or the drive itself from the MDFS without pressing the **Release Discs** button first. This form of error message is due to be phased out.

## B.2 File Server Messages & Internal Errors

### B.2.1 Non-fatal Errors

These are errors caused by some malfunction in the File Server or on one of its discs. Possibilities are:

Errors caused by some software problem in the File Server Code:

```
Task <nn> Error <letter> at address nnnn Bank bb
Task <nn> killed by timeout
```

Errors caused by corrupted discs, printed as each disc is booted:

```
Block 0 corrupt on Drive <drive_letter>
```

```
Block allocated twice, Drive <drive_letter>, block nnnn
Bad block number, Drive <drive_letter>, block nnnn
```

```
Bad backpointer on drive <drive_letter> in ....
Wrong no. of files in Dir. in ....
```

(These last two will cause the disc to go Drive read only)

```
2nd disc called <discname> in drive <drive_letter>
```

Printed while the File Server is On line:

```
Block deallocated twice, Drive <drive_letter>, block nnnn
```

### B.2.2 Errors accompanied by the System Error LED.

These are accompanied by the message File Server internal error:.

ABEND 0B01	Bad SCSI bus status from a SCSI interrupt. Can be caused by a bad SCSI data cable.
ABEND 0B03	Non-zero MESSAGE IN byte
ABEND 0B04	Unrecognised STATUS byte
ABEND 0B05	STATUS byte = 8 (BUSY)
ABEND 0B06	Error zero in non-extended SENSE DATA
ABEND 0B07	MESSAGE OUT status
ABEND 0B08	Bad DEVICE/LUN in an IDENTIFY during RESELECT

ABEND 0B09	Received a MESSAGE REJECT byte!
ABEND 0B10	Bad SCSI ptr
ABEND 0B11	SCSI_N <> 0 during RESELECT (loss of BSY?)
ABEND 0C01	Cache entry valid, clean and on disc chain!
ABEND 0C02	Bad logical disc number
ABEND 0C03	Writing to non-valid cache entry
ABEND 0C04	Cache entry corrupt
ABEND 0C05	Cache entry matched in TSCOMP
ABEND 0Fxx	Bad status during SCSI interrupt (xx = bus status)
ABEND 1101	Logical block no out of range in F_BLK

### **B.2.3 Flashing SYSTEM ERROR LED after powering on an MDFS.**

Look at the flashing carefully: it is repetitively a long flash followed by a nubor of short flashes. Count the number of short flashes and consult the table below:

1 flash	CMOS battery-backed RAM inconsistent. Follow the procedure for setting the station number to 254 to cure this problem (section 7.4).
2 flashes	RAM fault
3 flashes	SIO fault
4 flashes	WD1793 fault
5 flashes	CTC fault
6 flashes	SCSI bus fault

Make sure your winchester drive is plugged in correctly and powered. If you power-up the MDFS before the winchester you will get this error.

Two, three, four or five flashes indicate a fault which is not user-repairable. Please inform SJ Research.

## B.3 Winchester Disc Errors

### B.3.1 RODIME RO752/652 Winchester Disc Error Codes

Code	Sense Key	Meaning
03	4	Write fault. Power supply voltage out-of-limits?
04	4	Not ready (FATAL)
06	4	Track 000 not found
10	3	ID ECC error
11	3	Uncorrectable data error
12	3	No ID address mark
13	3	No data address mark
14	3	Sector not found
15	3	Seek error
17	1	Recovered read error by retries
18	1	Recovered read error by ECC
1A	5	Parameter overrun
1C	3	Error while accessing defect list
20	5	Invalid command
21	5	Illegal disc address
22	5	Illegal function
24	5	Illegal bit or byte in CDB. Can be caused by bad ribbon cable.
25	5	Invalid LUN
26	5	Illegal bit or byte in parameter list. Causes as Error 24.
31	3	Format operation failed
32	3	No spare location available
40	4	RAM diagnostic failure
44	4	ROM diagnostic failure
80	4	DC motor failed to start
81	4	DC motor speed error +/-1%
82	4	DC motor speed error +/-5%
83	4	Index calibration failed
Ex	x	Sense key = x, error code = 0
FD	-	Unknown status byte
FE	-	Device busy (i.e. executing command)
FF	-	Either RSS error or Bus jammed or No BSY.

### B.3.2 ADAPTEC ACB 4000A/4070 Winchester Disc Controller Error Codes

N.B. The Adaptec controller puts a 'Address Valid' bit in bit 7 of the error code. Hence error 91 = 80 + 11 which means error 11 with the address valid condition set to true. The Adaptec controller does not support Sense Keys.

#### Code Error

- 01 No index or sector signal found during read, write or format.
- 02 Seek complete signal not received from drive.
- 03 Write fault. Drive detected failure which disallows writes.  
Can be caused by power supply fault.
- 04 Drive not ready. Drive not connected or no power to drive.
- 06 Track 000 signal not received from drive.
- 10 ID field CRC error. Formatting information gone corrupt.
- 11 Uncorrectable data error. Data could not be recovered by retry or correction.
- 12 ID address mark not found. See error 10.
- 14 Record not found. Could not seek to track with correct ID.
- 18 Data check in no retry mode. See send diagnostic command.
- 19 ECC error during verify. Sector had bad data CRC.
- 1A Interleave error. Interleave is greater than the number of sectors per track on disk.
- 1C Unformatted disc or corrupt disc descriptor sector. The disc drive parameters (number of heads etc) will have to be re-entered when re-formatting the drive. (Z option in FORMAT)
- 20 Illegal Command. Command code is invalid or not implemented.
- 21 Illegal Block Address. Sector number out of range. Corrupt directory?
- 23 Volume overflow. Silly parameters to FORMAT, or number of blocks fields too large.
- 24 Bad Argument. Reserved bit not zero, invalid parameter or bad block list in the wrong order. Can be caused by bad ribbon cable.
- 25 Invalid LUN. Drive number greater than 1 addressed.
- 28 Cartridge changed. A disk drive cartridge was installed since the last time a command was executed. Can't see how this should happen!
- 2C Error count overflow. Posted when error count exceeds specified threshold.

Ex,FD,FE,FF as RODIME (see section B.3.1).

## B.4 SCSI Sense Key Definitions

SENSE KEYS apply to most SCSI devices, and are an attempt to give an idea as to what sort of error a particular error number refers to. Actual error numbers are technically vendor unique, although there are many conventional error numbers already in use. Currently, SENSE KEY information is only printed in Utility Mode with the debug mode enabled.

Sense Key	Definition
00	NO SENSE
01	RECOVERED ERROR
02	NOT READY
03	MEDIUM ERROR
04	HARDWARE ERROR
05	ILLEGAL REQUEST
06	UNIT ATTENTION
07	DATA PROTECT
08	BLANK CHECK
09	Vendor unique
0A	COPY ABORTED
0B	ABORTED COMMAND
0C	EQUAL
0D	VOLUME OVERFLOW
0E	MISCOMPARE
0F	Reserved
FF	SENSE KEY facility not supported (Adaptec will give this)

## B.5 Tape errors

Code	Sense Key	Meaning
08	4	Drive Communication Error. Can be caused if the tape is removed while it is being accessed.
10	3	ID CRC error. Badly Formatted tape, excessive tape dropout, or high external noise (RFI).
11	3	Unrecoverable Read error. Causes as Tape error 10.
15	3	Seek error. Causes as Tape error 10.
19	3	Defect List error. Tape might require de-gaussing and re-formatting.
21	5	Illegal Logical Block Address.
24	5	Illegal Bit or Byte in Command Block. Can be caused by bad ribbon cable.
27	7	Write protected. Remove tape and move the black tab.
28	6	New Cartridge Ready for Use.
29	6	SCSI RESET has occurred.
42	4	Power-on Diagnostic failure
A0	4	Background Noise error.
A7	4	Autoload failure. Try reinserting the tape.
A8	2	Cartridge autoloading. Wait for the drive to stop winding.
B0	2	No Cartridge in the drive.

Ex,FD,FE,FF as RODIME (see section B.3.1).

Flashing red LED after *autoload*: this is much the same as tape error A7.

### Errors generated by MDFS in relation to Tapes and Winchester:

No BSY Drive failed to select within the given time.

Drive either not connected or still winding tape while operation attempted.

RSS error Error during the 'get last error code' command. Can be caused by bad ribbon cable.

Bus jammed One of the eight control signals on the SCSI bus was being driven while the MDFS was trying to take control of the bus. Try powering off the system.

### Other errors not specific to tapes but which may occur during Tape Backup:

TOO MANY ERRORS More than about 20 errors of one sort or another have occurred.

Data fail @ nnnn Indicates that the data read from the tape did not match the corresponding data on the disc. This is caused by either a software bug or a hardware failure, or by a method described under *Check tape* in section 8.2.5 (Other operations). This error does not

indicate that there is necessarily anything wrong with the tape drive or winchester, or that there is a bad sector on either of them. It can be caused by a faulty ribbon cable, or faulty bus termination (see section C.3).

Leading/Trailing error Indicates that the tape has got out of sync. with the disc. This has been caused by running \*FAST on a version of ANFS pre 4.25.

## B.6 Floppy Disc Errors

Errors 08, 10 and 18 indicate that there is some problem in reading or writing the data on the disc. There are many causes of this. You can find that some discs will only read in some drives, or that a disc formatted in one particular drive will only work in that drive. Otherwise the disc may have developed a bad spot on one of its surfaces, or the drive read/write head may have become dirty. We recommend that if you get these sort of errors you try using the **Verify disc** option in Utility Mode in order to get a better picture of what is wrong.

Code	Error	Meaning
08	Data CRC error	Can't read the data off the disc. Try using another drive.
10	Sector not found	Can be caused by a step-rate which is too fast. Otherwise the disc may need re-formatting before re-use.
18	ID CRC error	Disc probably needs reformatting.
40	Write protected	You have added a write protect tab or changed discs without telling the File Server.
80	Not Ready	Hardware problem in the MDFS.
81	Disc timeout	Usually caused by someone 'illegally' removing a disc.
82	Trk 00 not found	Probably caused by a step-rate which is too fast.



# Appendix C: Installing a File Server

---

## C.1 Installing the Modular Disc File Server

The unit should be installed in a reasonably dust-free environment for the protection of floppy discs, preferably in a small room (for example a preparation or store room) near to the main computer classroom(s). A shelf or work-top of depth at least 400mm (16 inches) is best for the unit - this includes the depth of the File Server unit and also about 70mm for connectors on the rear. Take care to leave the ventilation slots clear under the unit and in the top, as overheating could otherwise occur.

It is recommended that wherever possible a special mains socket is fitted to run any File Server unit. This socket should be unswitched, and placed behind the unit or otherwise out of the way of accidental unplugging. If the computer area has a master switch, the File Server supply should be *independent* of it.

If you are using floppy disc drives connect one or more of them to the back of the unit. A maximum of 4 drives (normally two twin units) can be connected - one pair to each disc drive connector. If you possess a unit containing four drives, this should be connected to the socket marked DRIVES A AND B.

The disc drives should be set to *80 track* if they are switchable. Preferably 40/80 track switches should be removed, or at least taped into the 80 track position. Twin disc drives should be set up as for a BBC Microcomputer, with the lowest numbered drive select (DS) link (either DS0 or DS1) set for the top (or left-hand) drive, and the next numbered link (DS1 or DS2) for the bottom (or right-hand) one.

We cannot stress too strongly the need to use the best quality magnetic media and drives for a File Server, since in this application a considerable number of people will be affected by a disc failure.

Up to four hard disc drives can also be fitted to the MDFS (see §C.3 below). They should be plugged into the socket marked SCSI Bus Connector, on the rear of the unit.

Connect the printer(s) as described below. Connect the socket marked ECONET to the network with the lead provided. Connect the unit to the mains supply.

A BBC Microcomputer, or serial terminal, should be available fairly close by, so that operations using Utility Mode (see Chapter 7) can be done without too much walking. If you have a FAST EPROM (supplied with SJ Winchesters) it should be fitted to it as follows:

Remove the four screws, from the BBC microcomputer, securing the lid to the case; there are two at the back and two (with the larger size of head) underneath the keyboard. Lift off the lid. Remove the two or three nuts and bolts securing the keyboard, and pull the keyboard forward (there is no need to unplug it). Insert the EPROM, with the end containing a semicircular depression pointing toward the back of the computer. The BASIC ROM, which will be marked with the numbers PB01 or PB05 after the type number, should be plugged in to the right of the FAST EPROM. Reassemble the computer, and check that the EPROM works by typing \*HELP; the response should include **Fast terminal ROM ver n.nn**.

### C.1.1 Indicator lights

There are eight lights on the front of the MDFS unit.

#### **POWER** (green)

Indicates that there is a 5 volt supply to the MDFS microcomputer board. If this light does not come on when the unit is turned on, check the mains supply to the unit and the fuse in the 13 amp plug. If there is power at the socket and the fuse is OK but the light is still out, contact your dealer or SJ Research.

<b>ON LINE</b> (green)	Indicates that the File Server program is running correctly, and the File Server is in normal operation mode. This light will flash when the RELEASE DISCS button has been pressed to change the floppy discs in the File Server.
<b>UTILITY MODE</b> (yellow)	Indicates that the unit is in Utility Mode. If this light flashes, the system is waiting to enter Utility Mode, i.e. for a serial terminal, or a station running the FAST program to connect to the File Server. Pressing the RELEASE DISCS button in Utility Mode will boot the File Server.
<b>DISCS FREE</b> (yellow)	Comes on when it is safe to remove disc(s) from the system. Do not remove a disc in normal mode without pressing the RELEASE DISCS button and waiting for this light to come on. In Utility Mode this light will come on automatically when it is safe to change discs. The DISCS FREE light flashes when a File Server program is needed, and flashes more rapidly while the system loads the program.
<b>SERIAL PRINTING</b> (yellow)	Indicates that output is waiting to be printed on the serial printer; this may be users' or system output. This light will flash if the File Server printer buffer is full of system messages, so that the File Server cannot run until a suitable printer is available to print these out.
<b>PARALLEL PRINTING</b> (yellow)	Indicates that output is waiting to be printed on the parallel printer; this may be users' or system output. This light will flash if the File Server printer buffer is full of system messages, so that the File Server cannot run until a suitable printer is available to print these out.
<b>NO CLOCK</b> (red)	Comes on if the Econet clock signal is not present at the network connector on the rear: most commonly the MDFS will have been unplugged from the network, or someone has unplugged the clock box itself. If the clock box is connected and working, unplug it from the network and connect it directly to the File Server unit only, and check that the NO CLOCK light goes out. If it does not, there is a hardware fault either in the clock box, the MDFS or the connecting cable.
<b>SYSTEM ERROR</b> (red)	Lights when there is something wrong with the File Server. If the light comes on steadily, an explanatory message will be given on the system message printer. If this light flashes at power on, it indicates that some internal hardware device has failed (see appendix B).

## C.2 Connection of Printers

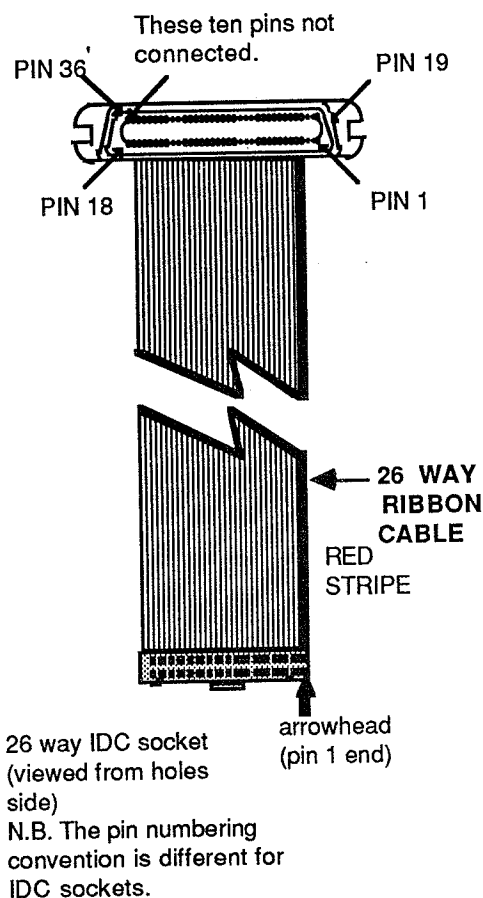
Two printer ports are supplied on the MDFS, so that a Centronics type parallel printer and a RS232 type serial unit may be connected.

The parallel printer is connected via a 26 way 'insulation displacement' socket and matching ribbon cable, to a suitable mating connector (usually a 36 way Amphenol type 57F-30360) for the printer. The lead and connections are exactly the same as those for a BBC Microcomputer (shown in Fig 1 below).

The serial printer also uses the same connector as on a BBC Microcomputer. This plug may unfortunately be inserted either way into the socket - no damage will be done, but the printer will not function. The socket is the same way up as on the BBC Microcomputer, so if the user subscribes to the convention of marking the top of the connector with a spirit marker, this will also hold good for the MDFS. The serial connections are given in Fig 2, along with the connections to be made to the industry standard D-type connector which will probably be fitted to a serial printer. The Baud rate for a serial printer is set up from Utility Mode, as described in Section 7.3.2.

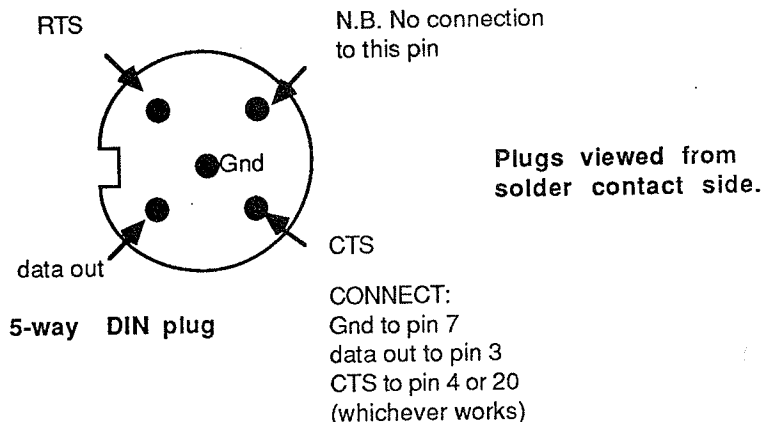
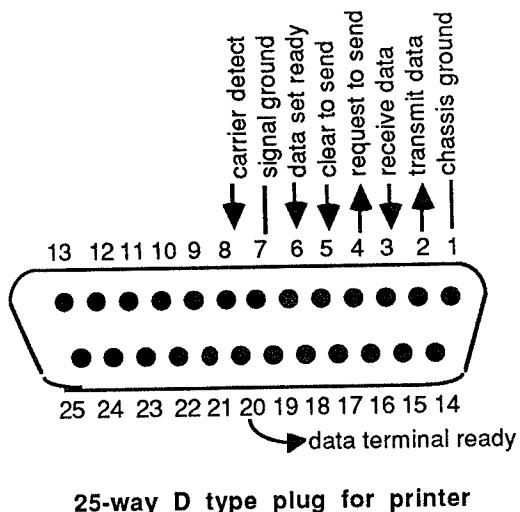
Many printers (especially Epson types) have an internal buffer, and a facility to remain 'busy' (after the buffer is filled) until a certain number of characters has been printed. This number of characters can be varied by setting internal links - it should be set to its minimum value for use with any printer server.

Printer leads may be bought from most computer dealers or from component distributors such as RS Components, or suitable made-up cables can be obtained from SJ Research.



**Fig.1 Parallel Printer Connections**

Signal pin No.	Return pin No.	Signal	Direction	Description
1	19	STROBE	In	STROBE pulse to read data in. Pulse width must be more than 0.5ns at receiving terminal. The signal level is normally "HIGH"; read-in of data is performed at the LOW level of this signal.
2	20	DATA 1	In	These signals represent information of the 1st to 8th bits of parallel data respectively. Each signal is at "HIGH" level when data is logical "1" and "LOW" when logical "0".
3	21	DATA 2	In	
4	22	DATA 3	In	
5	23	DATA 4	In	
6	24	DATA 5	In	
7	25	DATA 6	In	
8	26	DATA 7	In	
9	27	DATA 8	In	
10	28	ACKNLG	Out	Approx. 5ns pulse. "LOW" indicates that data has been received and that the printer is ready to accept other data.
11	29	BUSY	Out	A "HIGH" signal indicates that the printer cannot receive data. The signal becomes "HIGH" in the following cases; 1. During data entry 2. During printing operation 3. In OFF-LINE state 4. During printer error status
12	--	PE	Out	A "HIGH" signal indicates that the printer is out of paper.
13	--	SLOT	Out	This signal indicates that the printer is in the selected state.
14	--	OV	--	
15	--	NC	--	Not used.
16	--	OV	--	Logic GND Level.
17	--	CHASSIS -GND	--	Printer chassis GND. In the printer, the chassis GND and the logic GND are isolated from each other.
18	--	+5V	--	For CENTRONICS use only.
19 to 26	--	GND	--	RETURN signal GND level.



**Fig. 2 Serial Printer Connections**  
Issue 21 Apr 1987

## C.3 Installation of Winchesters and Tape Drives

### C.3.1 General Notes

Winchesters and Tape drives are all connected to the MDFS SCSI bus connector. Additional devices are connected by *daisy-chaining*, i.e. plugging in a new device into the connector of the previous device.

**You are advised not to connect or disconnect drives with the power applied.**

The SCSI bus (*bus* meaning a collection of wires or a cable) allows up to seven devices to be connected together and for any two to communicate with each other. On the MDFS communication is between the MDFS and one of the other devices, e.g. a disc controller. If there is more than one device connected to the MDFS, there needs to be a way of deciding which device the MDFS wishes to talk to. This is done by allocating each device a **unique** controller number in the range 0 to 7. All SCSI devices have a means of setting this number, although the procedure for doing this is manufacturer dependant: details are given below. On the MDFS, controllers 0 to 3 are allocated to disc controllers, controller 4 is allocated to the tape drive, controllers 5 and 6 are reserved and controller 7 is the MDFS itself.

The second characteristic of the SCSI bus is that it has two ends. Electronic signals do not like ends because the signals tend to bounce back when they meet an end. They then travel to the other end of the bus and bounce back again. Eventually they lose all their energy and the bouncing (technically known as *ringing*) dies out. The consequences of the ringing is sometimes to cause devices 'listening' to the bus to miss-read it, causing all sorts of problems. In order to prevent the ringing we need a *terminator* at each end of the bus. The MDFS already has a terminator in it, and therefore must be at one end of the bus. If you have a single winchester it must also have a terminator in it. This is usually not a problem because most drives come with terminators fitted as standard.

The problem comes when you want more than one device on the SCSI bus, such as a tape drive or a second winchester disc controller. In this case you will have to remove the terminators from the device or devices which are 'in the middle' (i.e. not at the ends), otherwise you would have more than two sets of terminators on the bus which will overload it.

Because each controller number must be unique, it follows that if you take two MDFS systems, each with a disc drive set up as Controller 0 (the default), you cannot plug both disc drives into the same MDFS without changing the controller number of one of the drives (to 1, for instance). You will also have to remove the terminators from one of the drives; these you will have to re-insert if you wish to connect the drives back onto the original MDFS, but you can leave the controller number set to 1, since that will not produce a conflict.

#### Further note to users of Adaptec Disc Controllers

It is also possible to use two drives connected to a single disc controller. As it is only the controller that is connected to the SCSI bus, one can add the second disc to the controller without changing the device number on the controller. However, you will have to change the device number on the disc for much the same reasons. See the drive manufacturer's instructions for how to do this.

### C.3.2 Setting the Controller Number and Terminators

#### For the RODIME RO752:

If the drive is in a case, you will need to remove it. On the side of the drive with all the components on it you will find a double row of pins marked S1, and a blue plastic link over one set of the eight pairs of pins. The pair nearest to the mark S1 should be made to set the number to 0, the next pair only to set it to 1, the next pair up for 2 etc.

The terminators are the three long thin yellow things (they are called *resistor packs*) marked SIL1, SIL2 and SIL3 near to the 50-way connector on the drive. Take all three of them out and put them in a safe place: taping them to the inside of the case is a good idea. If you need to put them back in, with the word **SIL1** the right way up, the 'dot' on the resistor pack should be to the left.

### For the Adaptec ACB4000A Controller:

You will need to identify the controller board as distinct from the drive itself, and then remove screws etc as necessary so that you can get to the component side of the controller board. You will see a double row of pins marked **J5**. The pin-pairs marked **A B**, **C D** and **E F** are used to select the controller number in binary. With all three pairs *open* (i.e. not connected) number 0 is generated. To make controller 1 you need the binary number 001, so that only the **A B** pair should be connected. To make the link you will need a molex connector which you can pinch from a BBC microcomputer's station number selector.

The terminators are two long thin things marked **RP3** and **RP4**. Store them in a safe place if you need to remove them, or if you need to put them back in again, the 'dot' should go nearest the big 50-way connector. **Warning:** there are some boards which have the terminators soldered in, and unless you feel very competent you should remove the terminators from another controller board instead.

### C.3.3 Installing Tape Drives

These notes apply to a tape drive supplied in a plastic case.

The tape drive has three connectors on it: two 50-way data connectors and a power connector. Plug the power plug into one of the two **Power Out** sockets on the back of the MDFS. Plug the 'flying' end of the 50-way tape cable into the MDFS, where the hard disc used to plug in. Plug the 'flying' end of the hard-disc box into the 50-way plug on the back of the tape. You power the hard-disc as before.

**The Tape Drive should go in between the Winchester and the MDFS.**

This is because the Tape Drive comes with no terminators installed. This means that you can remove the tape drive for use on another site by plugging the hard-disc back into the MDFS. When re-connecting the tape drive do not plug it straight into the back of the hard-disc.

### Copyright.

Use of the tape software is restricted to use on Tape Drives purchased from SJ Research, as laid down in our terms and conditions.

### C.3.4 Installing SJ Research Winchesters

SJ Research winchesters come ready formatted, with a copy of the standard release software and File Server code already mounted on them. To connect to an MDFS, first make sure that power is off, and then put the expansion case either above or below the MDFS. The power lead should be connected to the D-type outlet on the back of the MDFS **and the screws fastened**. This is important because without this fastening the connector can work its way out and the voltage to the expansion case can become intermittent, resulting in loss of data. Next connect the 50-way lead into the **SCSI Bus Connector** on the back of the MDFS (it will only go in one way round). At this stage, disconnect any floppy discs.

Power the MDFS on; you should notice the disc LED flash on very briefly and hear the drive motor speeding up. After a while the MDFS **Disc Free** LED will start to flash; press the **Release Discs** button on the MDFS and after about 4 seconds the disc LED will come on again while the MDFS searches for the File Server program. It should load this and turn on the **On Line** LED. You should log-on with **\*I AM SYST SYST** and check basic operation of the File Server. Ascertain the version of the new File Server code, by typing **\*VERS**.

The next thing to do is to assign a suitable name to the disc. It will currently be called **MASTER** or **HARD 1**. If it is called the former then you **must** change it. To change the discname go into utility mode and use the **RENAME** option.

The rest of the procedure need only be followed if you have already used the MDFS with floppy discs; it ensures that you have a backup copy of the new File Server code.

## Making a backup copy of the File Server code.

Press the button, and power off. Reconnect a floppy disc drive and re-boot the File Server, using the disc with the File Server code on it that you normally use, log-on and find out the version number of the File Server code. If the version on the winchester is greater than that on the floppy, you will need to make a copy of the new version onto a floppy disc as a backup: see §4.4 for how to do this. If the versions are the same then you already have a backup, and if the winchester version is less than your usual version then we have supplied the wrong version: please inform us!

## What to do if you get a flashing System Error LED

Six short flashes followed by one long flash indicates that something is wrong with the SCSI Bus. This could be because the data cable is plugged-in the wrong way round (although the lead should be polarized to prevent this) or that there is no power to the drive. MDFS ROMs previous to version 0.96 had a problem (with identical symptoms) when used with certain batches of disc drive, so please inform SJ Research if you have such a ROM.

## Some technical information

A single winchester will come set up as controller 0, and with terminators fitted. The case has internal connections for a second winchester, which, if subsequently fitted, should be set to controller 1 and have the terminators removed. See the relevant sections in this chapter.

## C.3.5 Installing BBC-Compatible Winchester

We can summarize the operations needed for the installation into 4 parts:

- a) Copying any existing data off the disc onto some other media
- b) Removing some redundant parts and adding a new cable to the winchester disc unit in order to connect it to the MDFS
- c) Re-formatting the winchester
- d) Copying on the File Server code
- e) Copying back any data transferred off in a)

Clearly steps a) and e) are only relevant if the disc has any interesting information on it already. Methods of doing this all depend on what other system you intend to use for temporary storage. If you have winchester in Acorn Level III File Server format, SJ Research will, for a fee, perform the whole conversion operation for you, otherwise you can use the utilities **Archive** and **Getback** to transfer the data via floppy discs.

## Connecting the Winchester to the MDFS.

Take the cover off the winchester box. The 34-way ribbon cable that used to connect the BBC microcomputer and the winchester either plugs into the back of the box or goes directly inside the box. Either way, identify the 34-way cable that is inside the box. It should lead to a small board. Remove this board and all the 34-way cabling. The little board should also be connected to the Adaptec controller board via a short 50-way cable, or it may be plugged straight in. You won't need the short piece of 50-way cable either.

You will now need to make-up or buy (from your local computer store) another 50-way ribbon cable at least 1 metre long. This lead should have an **IDC receptacle** (i.e. female connector) at both ends. The lead should then be connected to the 50-way plug on the controller board, fed out of the box and the other end connected to the MDFS SCSI Bus Connector. Pin 1 on the Adaptec board is at the end nearest the red LED, and on the MDFS it is marked with a dash. When powering up the system you will need to apply power to the disc drive before the MDFS (or you can turn them on simultaneously) otherwise you will get a flashing System Error LED.

## Re-Formatting the Winchester.

You will need to get into Utility Mode in order to use the format program. Connect a floppy disc drive to the MDFS, insert the master release disc, and switch on the MDFS. Refer to section 7.2 for further details, bearing in mind that you will not have a copy of the \*FAST ROM. If you have Utility Mode version 1.00 or greater, then you can use the **List Discs** option to check that your winchester is connected properly.

For Example:

LIST DISCS

Discs currently available:

A: Name: Master      size: 800K  
E: Not an FS format disc.

or

E: Disc error in root - bad disc.

Do not worry about these two messages, as they are quite normal before you have formatted the disc.

However, if the Utility Mode version is less than 1.00 it will almost certainly hang up if you try the **List Discs** option. The version number is printed at the top of the main menu.

If the winchester has already been formatted (e.g. for use on a BBC microcomputer) it should be re-formatted using the 'B' (Acorn) option in **FORMAT**. This option applies to all pre-formatted drives even if they were not purchased from Acorn. If the drive has never been formatted, or you get **Error 1C**, you will need to use the 'Z' option, referring to the drive manual for details of the number of Heads, Cylinders etc. For further information on formatting, see §7.3.6.

You should check the drive for a *defect list*, which should be stuck to the drive itself. This list should have numbers of the form xxx-y-zzzz (e.g. 110-3-2305), which list the locations of any *media defects* (imperfections on the surface of the discs). The disc controller must be informed of these defects so that it can avoid using them for any user data. If you have such a list, you should reply 'Y' to the prompt

Enter defect disc (Y/N)

and enter the numbers as appropriate. Do not loose the defect list, as you will need it if you have to format the winchester again, which will necessary if you wish to enter additional defects. Sectors which go bad during the life of the drive will have to be entered in this way.

After the format operation has finished (which takes approx. 5 minutes) the MDFS will write a header to the disc and then verify all sectors on the disc. You can use the **List Discs** command (on any version of the Utility Mode software!) and use your winchester as a normal File Server disc.

## Copying the File Server Code onto the Winchester

See §4.4 for details.

# MDFS software version 0.AA Release Notes

## New features

This release supports :

- Partitioning of large hard discs into 35 Mbyte logical discs.
- Up to 4 hard discs or partitions (formerly 2).
- Retrieval of individual files from tape.
- 2048 accounts (numbered 000 to 7FF). See separate notes.

## Important change

It is important to *park* the heads of hard discs if the equipment is to be moved. Previous versions of the fileserver parked the heads whenever the release discs button was pressed. Some modern disc drives will shut down completely when instructed to park the head, which is undesirable when simply changing floppy discs.

The new software will only park the heads if the release discs button is *pressed and held for five seconds* when the fileserver is running. This will cause the discs free lamp to illuminate steadily, indicating that the fileserver is completely shut down. When changing discs, the button should just be pressed momentarily, which will give the usual flashing discs free/online lamps.

Alternatively, the heads may be parked with the **Z** command in utility mode.

You are warned that transporting hard disc drives without having parked the heads can cause **permanent damage** to the surface of the disc.

## Partitioning of large discs

Hard discs with a capacity greater than 35 Mbytes are now divided into *partitions*, each of which functions as if it were a separate disc drive. Each partition has a capacity of 35 Mbytes, except the last partition on each disc which will use up any remaining space on the disc. For example, a 95 Mbyte disc will have two partitions of 35 Mbytes plus one partition of 25 Mbytes. The fileserver can only support a total of 4 partitions when online. If discs with a total of more than 4 partitions are connected, the fileserver will use the first partition on each disc, plus the largest partitions available from the other discs up to the maximum of 4. Hence with a 95 Mbyte disc (35+35+25 partitions) plus a 37 Mbyte disc (35+2) the 2 Mbyte partition will be ignored. Four floppy discs can always be supported in addition to any hard discs.

When the fileserver is online, the separate partitions function as independent discs and are referenced by name. Use \*FREE to list the available discs/partitions.

In utility mode, the discs are referred to by letter (corresponding to the hardware controller number & drive number). If a large disc is selected, the system will prompt for a partition number (key 1 for the first partition, 2 for the second, up to the number of partitions on that particular disc). In the case of the Verify command, individual partitions can be verified or the whole disc can be verified at once by specifying a partition number of zero. Use the **L** (list discs) command to see the partitions available on all discs.

When using a tape drive, each tape can hold the contents of one disc partition. It is therefore useful to arrange your files on the disc such that frequently updated files are in one partition which can be backed up daily, leaving constant material (such as software packages, archive material, read-only databases etc.) on another partition to be backed up less frequently.

## Initialising existing hard discs

Customers with discs larger than 35 Mbyte which have been used with earlier versions of the fileserver will only have been able to use the first partition on the disc. When the new software is installed, the other partitions will still not be available as the root directories will not have been created when the disc was formatted - messages such as *Block 0 corrupt on drive E2* will be produced on the printer.

To create these directories, either the disc must be formatted again, or the roots of each partition must be initialised. The format command now prompts for separate disc names for each partition. Beware that formatting a disc *erases all data from all partitions on that disc*. Only format a disc if you are happy to lose the data stored on that disc.

To initialise a partition without corrupting other data on that disc, the **I** (initialise disc) command may be used (in utility mode). Note that this command does not appear on the menu which is displayed, but works just like the commands that do appear on the menu. Before using the **I** command, type **L** to obtain a list of the discs and partitions available. Partitions which need initialising will say *Not a filesaver disc* in place of the disc name - on discs formatted with the old software, partition 1 will contain the existing data, while partitions 2, 3 etc will need initialising. Now type **I** and enter the drive letter followed by the partition number. The current name of the disc will be displayed (this should say *Not a filesaver disc* if the partition needs initialising), and the program will ask for a name for that partition. Enter a name which is different from any other discs or partitions in your system. Finally you will be asked to type **Y** or **N** to confirm that the details are correct - type **Y** if you are sure that you have specified the correct partition. Take care **not** to initialise partition 1 as that would erase the data previously stored on the disc. Repeat the **I** command if there are more partitions to initialise.

## Retrieval of individual files from tape

It is now possible to read backup tapes without restoring the whole tape onto a disc. This is particularly useful for recovering files which have accidentally been deleted.

The facility works while the filesaver is online, by making the tape appear as if it were a very slow disc drive. It is not possible to write to the tape in this mode.

If a tape is inserted in the tape drive while the filesaver is online, a special directory **%TAPE** becomes available. This is equivalent to the root (\$) directory on the disc that was backed up onto the tape. Hence the following might be used to recover a BASIC program :

```
>*I AM FRED
>*DIR %TAPE
>*DIR FORM3
>*DIR FRED
>*CAT
FRED          (073)      Owner
ARG1          Option 00 (Off)
Dir. FRED     Lib. LIBRARY

Bescfix      WR/r      BFASTCOMP  WR/wr      Bincode      WR/r      BMINITERM    WR/r
Bsafeterm    WR/r      Bxmit      WR/r      CARDS        D/         cst          D/

>LOAD"Bxmit"
>*DIR
>SAVE"OldXmit"
>*UNLOADTAPE
>
```

No privilege is required to access the tape; the files on tape still have account numbers and access letters attached, so access is controlled in just the same way as files on the main disc. All the usual commands (eg. **\*CAT**, **\*EX**, **\*INFO**) can be used to inspect the contents of the tape. The utilities *Copier* and *Multicopy* can be used to copy groups of files.

One problem with this system is the slow response of the tape drive, which can cause *No Reply* errors. If such an error occurs, wait for the tape to finish winding and repeat the sequence of commands. The necessary data should now be held in memory in the filesaver and so the commands will succeed immediately. The following guidelines will minimise the possibility of these errors :

- Choose a time when there are as few people as possible using the filesaver, as other users will use up valuable memory space.
- Note that the example above selected the directory in a series of steps, rather than **\*DIR %TAPE.FORM3.FRED** which would have required the filesaver to do all the work in the time allowed for one operation. Always divide up long pathnames in this way.
- Use *Copier* or *Multicopy* on a BBC Master (or ET or Compact). This combination allows longer for each operation.

Before removing the tape, it should be *unloaded* to protect the surface of the tape from contamination. To do this, either press the *Release Discs* button, or use the \*UNLOADTAPE command. When the tape has finished winding, it may be removed by pressing the large button under the tape slot.

# Fileserver software with 2048 accounts

## Introduction

The fileserver software for HDFS and MDFS has been enhanced to allow the use of 2048 account numbers, with effect from version 0.AA . The new system is upwards-compatible from the old; no special action is required when installing the new software onto existing discs, but once the enhanced facilities have been used on a disc, that disc should not be used with earlier versions of the fileserver.

A new version of EDITPASS is provided to allow access to these accounts. In addition, a suite of programs for managing very large password files is now available - see the separate documentation.

## Availability of account numbers

The new accounts are identical to the old in most respects; a separate balance can be kept for each account on each disc and files can be freely assigned to any account. The main difference is that there are now some restrictions on the way in which account ownership can be given to users.

Under the old system, users could be given ownership of any combination of accounts. However, each user would typically have one personal account in which to keep all their own files, while some users would share ownership of a few more accounts giving access to printers, or for shared project work. In addition, the system manager and other users in positions of authority would be granted ownership of other users' personal accounts for supervisory purposes.

The new system formalises this pattern of use. The system now records a personal account for each user, which may be freely chosen from the whole range of account numbers (000 to 7FF), but any shared accounts must have numbers between 000 and 0FF. For the system manager and other 'super users', it is possible to have access to more than one account above 0FF, but only in blocks of 64 accounts. For example, a class of pupils might have personal accounts 100, 101, 102 ... 11A. To give the class teacher ownership of the pupils' files, it would be necessary to give him ownership of accounts 100 to 13F. See below for the available blocks of account numbers.

The usual allocation of account numbers on a large system is to allocate all users' personal accounts between 100 and 7FF, starting each group of users on a multiple of 64 (100, 140, 180 etc.) so that super users can be given blocks of accounts which match up with the users that they are to supervise. Accounts 000 to 0FF are then available for shared use. Note that 000 is usually the account number of the root (\$) directory, and so should only be owned by the system manager. It is customary to reserve a few more accounts (say 000 to 01F), leaving the remainder up to 0FF for shared use.

There is no restriction on the use of personal account numbers, so on a small system personal accounts could start from, say, 020 and accounts above 0FF need not be used at all.

If the personal account is set to zero, ownership of account 000 is *not* granted, and the user effectively has no personal account. This is useful for public users, such as BOOT or ANONPRINT.

Note that when a user is printing, the job in the print queue is given an auxiliary account number equal to the user's personal account number, or if the user has no personal account equal to the highest numbered account that the user owns, or zero if the user owns no accounts at all. This is so that each user has ownership of his own print jobs, which is necessary to delete, reroute, or flush them.

## Storage of account balances

To store the current balance for all accounts requires 4K of disc space on each disc. For small systems where not all accounts are in use, particularly with floppy discs, this is an unnecessary overhead. The system therefore does not allocate this space initially, and so several account numbers share the same balance; account 000 has the same balance as 100, 200, 300 etc. If a large number of accounts are in use, the fileserver can be instructed to take up the necessary disc space and the accounts will have independent balances.

**\*MAXACC 7FF** Uses 4K of disc space, and all accounts have independent balances.

**\*MAXACC 1FF** Uses 1K of disc space, accounts 000-1FF all have independent balances, accounts 2xx, 3xx, 4xx, 5xx, 6xx, 7xx all share balances.

Note that \*MAXACC only applies to the currently selected disc; to reserve balances on all discs, it is necessary to select the disc with \*DIR (or \*SDISC) and then use \*MAXACC, repeating for all the discs on the system.

Users of hard disc systems would usually use \*MAXACC 7FF on all discs and use the accounts independently. With floppy discs, it is rarely necessary to have more than 256 accounts on each disc, but users on separate discs cannot share an account number as their files need to be protected from each other. If each floppy disc is allocated a range of accounts, say 110-1FF, 210-2FF etc., and other accounts are not used on that disc, each account will still have an independent balance (because balances are maintained separately on each disc) at no extra cost in disc space.

The effect of \*MAXACC is permanent; once allocated, the disc space cannot be recovered without re-formatting the disc.

### Upgrading from earlier versions of fileserver software

The new software should be installed in the usual way (*TPOKER* on HDFS, or by copying \$.FS on MDFS). In the case of MDFS, care should be taken to copy the new version of \$.FS to all discs which are used to start the fileserver - especially where floppy discs are used. The only change which will be apparent at this stage is that account numbers in \*INFO or \*EX now have three digits; no users (even *SYST*) will own the new accounts, so the system will continue to operate as before.

No action is required if the new accounts are not used; the new software operates identically to the old.

To start using the new accounts, it is necessary to use the new version of EDITPASS, initially to give *SYST* ownership of accounts 100-7FF. When existing users are inspected with the new EDITPASS they will have no personal account, but the account which they have been using as a personal account will appear as one of their shared accounts. There is no need to change this, as all users will have access to their files, but in due course it may be desirable to rationalise existing allocations of account numbers to fit in with the new scheme.

In most cases, it will be necessary to use \*MAXACC before allocating any new accounts to users. Note that when the account balances are separated by \*MAXACC they are not bankrupted, but each account will have the same balance as those which it was formerly sharing balance storage. Care must be taken to give new users an appropriate starting balance, or all the balances can be zeroed to avoid future mistakes :

```
10 FOR A%=&100 TO &7FF
20 OSCLI"DEBIT "+STR$~A%+" 65535"
30 NEXT
```

(Note that this requires BASIC 2 or better)

### Available blocks of high numbered accounts.

Users who need ownership of more than one account with numbers greater than 0FF may be given ownership of one or more of the following blocks. If an attempt is made to give ownership of a range which does not fit exactly onto these blocks, EDITPASS will allocate enough blocks to cover the whole of the specified range, which will give ownership of more accounts than were actually specified.

100-13F	140-17F	180-1BF	1C0-1FF	200-23F	240-27F	280-2BF	2C0-2FF
300-33F	340-37F	380-3BF	3C0-3FF	400-43F	440-47F	480-4BF	4C0-4FF
500-53F	540-57F	580-5BF	5C0-5FF	600-63F	640-67F	680-6BF	6C0-6FF
700-73F	740-77F	780-7BF	7C0-7FF				

### FDFS compatibility

The features of this new software will *not* be available on the FDFS. MDFS users who insert FDFS discs for data interchange should avoid using the new facilities on FDFS format discs. In particular, \*MAXACC should not be used, and when editing the password file personal accounts and account numbers greater than 0FF should not be used.

## Preliminary User Documentation for Password Management System

The new password file editor edits password files by converting them to a human readable ASCII text file which is then processable on any text editor. Additions and changes to the existing password file will be made by composing a text file in ASCII which is then merged with the text file that has been generated from the ASCII version of the %PASSWORDS file. This document outlines the format of these ascii password files.

There are two basic forms that these text files can take. The first is a file that contains details of modifications that are needed, eg adding new users or modifying existing entries, this is a "Mod-file". The second is a file which is generated from an existing password file, or from the merger of a mod-file and a file generated from an existing password file, this second type of file will never contain modification instructions only entries for users and is called the "Gen-file".

### **Format of data for individual entries**

The data about each user entry is held in two different ways, as data "Local" to that particular entry and as data "Global" to a range of entries (or the whole file). Particular information such as URD or account ownership is specified by assigning two a fixed keyword of which there are two types Global and Local. The Global keywords are as follows:

ACC	For normal accounts, and for blocks of personal accounts.
BASE	For the base of the URD to which the UID is added.
FLAG	Which contains the different possible flags as two letter symbols.
LIB	The library path.
PASS	The users password.
BOOT	A default boot option.
PACC	The user's personal account {Used only in Mod-files for setting start of search for free personal account numbers}

The Local keywords are:

ACC	As above.
URD	A full URD.
FLAG	As above.
LIB	As above.
PASS	As above.
BOOT	As above.
DEFAULT	Indicates the default user. In order to unset a current default user DEFAULT must be given the value "0".
PACC	Personal account. Set to "" when no personal account is required.

Assignments are made to keywords as follows:

Keyword = "data";

eg ACC = "1,2,3"; or BASE = "\$.form3";

Sometimes it is necessary to cancel the effect of a keyword or undefine it, this is done by:

Keyword = UNDEF;

eg PASS = UNDEF;

UNDEF must be in upper case.  
Note that there are no quotes

In most cases the data assigned to a keyword is obvious. For accounts ">" is used to indicate a range and "-" to indicate removal of particular accounts, and "+" for addition. {Note that "+" and "-" only take effect in modify mode}. For the FLAG keyword assignment data is in the form of two

letter combinations which are as follows:

Pw	Password locked.
Sy	System.
Ns	No short saves.
En	*Enable required.
Nl	No library.
Ro	Run only user.
X1	Reserved
X2	"

If an option is preceded by a "-" it is removed and by a "+" it is added to the current list of FLAGS in use. If neither "+" or "-" is given then the value of FLAG is used directly. {At present the use of "+" and "-" only work in "MODIFY" mode}.

Global and Local keywords are distinguished from one another by the use of { and }. These curly brackets follow the UID and all keywords contained within are taken as Local to that user. Thus the form for a user entry is:

Global assignments  
UID { local assignments }

Both the Global and Local assignments are optional, however the curly brackets must always be present. It is recommended that where the local assignments flow over one line that space is left under the UID so that it stands out on the page. However this is not a requirement. Thus a well formed user entry might look like this:

```
ACC="1,2,3"; BASE="$ .form3"; FLAG="SaEn";
```

```
ARG      { PASS="Wombat"; LIB="$ .SJLIB"; ACC="0>FF"; FLAG="Sy";  
          PACC="1FF"; BOOT="3"; }
```

{In the future the quotes will be optional except for assignment to the PASS keyword.}

There are a variety of different ways in which this entry might be used. These are termed "modes" and there are three different modes as follows:

#### .Add.

In this mode the user entries are taken as new users. If a user of this name already exists an error is generated. In the case of a correct addition a personal account is automatically assigned (unless the PACC keyword has been assigned to explicitly), and a set of instructions placed in the file "!mkdir", that can be \*EXEC'ed to create the directory structure for all the new users. If MERGE is used repeatedly on the same file this continues to add the commands to create new user directories to the existing !mkdir. Therefore it is important that the !mkdir is deleted after use.

#### .Remove.

The specified users are removed from the password file. Obviously no global assignments or local assignments are needed, however it is not an error for these to exist. This makes it possible to remove blocks of users and later restore them just by changing the mode information for those entries.

{ A future addition will be the option to remove the users files and generally tidy up the directories, this will be done as follows:

```
games      { REMOVE="1 ";  
}
```

## .Modify.

The data in the user entries is used to supplement or modify the data already held in an existing entry. It is an error for the user not to already exist. In this mode the undefining of a keyword is necessary. Because if a Global keyword is set it must be possible to undefine it, so as to allow subsequent user entries to retain their old values for this particular keyword. Local keywords may also be "UNDEF"ed to allow the current global to be ignored for that user.

These three modes only apply to the mod-file and will not be encountered in the gen-file. To select a particular mode a line is entered into the mod-file before the user entries to which it refers as follows:

.mode.

```
Global assignments
UID { local assignments }
UID { " " }
```

.mode.

```
UID { local assignments }
UID { " " }
```

The possible modes are:

.Add.  
.Remove.  
.Modify.

In the mod-file there is no default mode so one must be set before any UIDs are given, otherwise there will be an error. The different modes will cause different effects if an incomplete user entry is given, ie not all keywords are assigned either globally or locally. For Modify this implies that the data from an existing entry is to be used. For Remove no notice is taken of the data. For Add sensible defaults are assumed as follows:

ACC	No accounts unless PACC is in the range 0-FF.
URD	Taken as \$.<uid>
PASS	No password
LIB	Standard library.
BOOT	Taken to be off.
FLAG	Taken to be no flags set.
PACC	Assigned a sensible unique value.

## **Pseudo Modes**

There are further modes that are used to control the flow of processing. The first is ".END.", it is used in both Mod-files and Gen-files to indicate the end of the useful data in the file. Its use is optional. One use for it is to cause the remainder of a file to be ignored by one of the processing programs, though this is of little value in everyday use.

The second of these is .users xx. that informs the password file generator how many users there are to be in the file. This is inserted automatically in genfiles; the user need only be concerned with it if editing genfiles or other intermediate files directly rather than by merging in mod files.

## **Further notes**

Although a Local keyword exists for PACC it is assumed that personal accounts will be assigned sensible values automatically. This is possible because during the first part of the edit process, when the ASCII file is generated from the existing password file, a map of used personal accounts are

generated. From which suitable values are chosen from unused personal account numbers.

Because of the automatic generation of personal accounts there are a number of special effects that are generated by particular settings of the PACC keyword. If a global PACC is set then searching for personal accounts for new users will start from that value. This continues until it is unset by setting the Global PACC keyword to UNDEF, at which point the search for personal account numbers will resume at the lowest free personal account numbers. (Setting the Local PACC keyword to UNDEF will also switch the search, but only for that user) Personal accounts start at &100, however it is possible to set PACC to values below this, if this is done a warning is generated, but the assignment will take place. If a PACC is set to "" or to "0" then this is taken as meaning unset the personal account, again a warning is generated, and the personal account is unset. For newusers where the PACC has been explicitly set to "0" there is a problem in deciding which account to give to the URD, in this case the highest account number from the local ACC keyword is used, and in the case of there being no local ACC no access to the URD will be given to that user. (The reason that only the local keyword is scanned is so that account ownership of common accounts, such as PRINTQ may be given to a user with no file access).

Any line beginning with a "&" is taken as a comment and totally ignored.

### Program suite

The programs are run in the following order:

CHAIN "CONVERT"	This converts the %PASSWORDS file to the Gen-file
CHAIN "PARSE"	This parses the Mod-file
CHAIN "SORT"	This sorts the products of PARSE
CHAIN "MERGE"	This merges the Gen-file and the product of SORT
CHAIN "GENERATE"	This generates the new PASSWORDS file
*RENAME PASSWORDS %PASSWORDS	

The use of CONVERT is self explanatory, it will need to be used each time a modification is to be made to an existing password file. The PARSE program will be the most used, as this checks the user generated Mod-file syntactically, and produces appropriate error messages. Its output is placed in the TEMP sub-directory. This is a text file that has had all the GLOBAL assignments removed, and all the PACCS for new users inserted. This file is then sorted by SORT into alphabetic order ready for merging by MERGE. MERGE processes the Mod-file and Gen-file. It is at this stage that new users are checked for name clashes with existing users, and that users specified for modification actually exist. The final product of MERGE is a text file called "passtext".

MERGE also generates a file called "!makedir". This contains executable \*commands to generate the directory structures for the newusers created during the merge process. Finally if all has gone according to plan the GENERATE program is used to create the new password file, this is called "PASSWORDS" and is generated in the current working directory. In order to replace the existing %PASSWORDS with this new file the following command should be used:

```
*RENAME PASSWORDS %PASSWORDS
```

This is the only way that a new password file should be installed.

It is important to always run SORT after PARSE, and to run PARSE each time the Mod-file is changed. There is one important restriction on the size of the Mod-file, that is that they cannot contain more than 256 users, this should not present a problem as MERGE can be used repeatedly on the same file.

The whole process described above can take some time to complete, and it is envisaged that it will only be used when making large numbers of additions or modifications to a password file. For everyday usage the normal program "EDITPASS" should be used. Where there are too many users

for this program to work, or information about personal accounts needs to be changed then the program QEDIT must be used, this has a user interface very similar to EDITPASS, but only works on a single user at a time, thus not needing to store the password file in local RAM.

In order to install the suite of programs there is an executable file called "install" that creates a suitable directory called "pwmanage" in which the programs can reside.

### Further technical information

Since all of the passwords are displayed in ASCII text in the files, it is very important that only the system manager has access to them, and they should be treated with as much secure respect as %PASSWORDS itself. Each of the programs does its best to stop unauthorised people being able to see the files, but as always security is only as good as the system manager. It is intended that in future releases of the software passwords will be encrypted.

All of the programs can run on DFS rather than network, however the variable "net%" in each program must be made FALSE for this to work correctly. This variable along with other "tweaks" can be found after the comments in the first few lines of the programs. In order to suppress all output, except errors and warnings, the variable "test%" must be set to FALSE. Should there be any need to change the filenames used by each program, these can also be found assigned to string variables within the first few lines of the code in each program. If these are changed it is important to change the names in all of the programs.

### Formal definition of file spec

```

<file> ::= <gen-file> | <mod-file>
<gen-file> ::= [<userdata>] | [<global assignment>] [<userdata>]
<mod-file> ::= .<mode>. <gen-file>
<mode> ::= Add | Modify | Remove | End
<global assignment> ::= <global keyword> = "<keyword value>";
<userdata> ::= <Uid> | <Uid> { [<local assignment>] }
<local assignment> ::= <local keyword> = "<keyword value>";
<Uid> ::= [<alphanum>]
<global keyword> ::= ACC | LIB | PASS | BOOT | BASE | FLAG
<local keyword> ::= ACC | LIB | PASS | BOOT | URD | FLAG | PACC | DEFAULT
<keyword value> ::= <acc> | <lib> | <pass> | <boot> | <urd> | <flag> | <pacc>
                     <default> | <base> | UNDEF
<acc> ::= [ <hex>, | <hex> '>' <hex>, | - <acc>, | + <acc> ]
<lib> ::= <path>
<pass> ::= <alphanum>
<boot> ::= 0 | 1 | 2 | 3
<urd> ::= <path>
<pacc> ::= <bighex>
<default> ::= 1 | 0
<base> ::= <path>
<flag> ::= [<flagsymbol> | +<flagsymbol> | -<flagsymbol>]
<flagsymbol> ::= Sy | Ns | Ro | Nl | En | Pa | X1 | X2
<path> ::= [<name>.] | $<discname>.<path>
<discname> ::= <alphanum>
<name> ::= <alphanum>
<hex> ::= <hexit> | <hexit><hexit>
<bighex> ::= <hex> | <hexit><hexit><hexit>
<hexit> ::= 0|1|2|3|4|5|6|7|8|9|A|B|C|D|E|F

```

Note there is no case sensitivity, as every alphanum is taken as upper case (?? UNDEF seems to be an exception). Forms shown are only preferred forms.

# S.J.U.G.

The S.J.U.G. was formed as a user group for people using S.J. Research file servers, to allow them to exchange information (possibly software), and to be in closer contact with S.J. themselves.

There is considerable duplication of effort in writing network utilities, finding bugs and adapting software to run on the network. We see S.J.U.G. as a medium for distributing this information along with users articles and experiences, software reviews (and compatibility), new ideas and imaginative uses for Econet as well as help for new users.

Do you have any ideas on areas you would like covered by S.J.U.G. or any experience you feel would be of benefit to others? Below is a short list of some ideas that will be covered.

Fast file-handling on Econet	Communications with a network via modem
Word processing on the net	Network mailbox
Music on the net	Adding *COMMANDS
Handling graphics dumps	Sideways RAM's and utilities
Which Econet utilities are worth buying	Problems with adapting disc software
Econet hardware faults	Licencing software
Lightening strikes and interfaces	Errors in manuals
Differences between NFS versions	Additional utilities
How best to utilise your printers	Local Viewdata
Problems with security	Incorporating Ceefax/Prestel
Multiple file copying to and from disc	Bridging networks
Swapping and selling software	Links to PC compatibles

We would like to stress that the S.J.U.G. is run independently from S.J. Research, however we are in close contact with them for information and comments. We do not intend to make a profit from running S.J.U.G., however we must cover our costs for printing, postage, phone bills and other ancilliary costs.

**We would ask you to join S.J.U.G., if you do wish to join please -**

1. Return this form - you will then be put on our mailing list so that you will receive the free copies of the newsletters, we currently publish three issues a year.
2. Enclose with the form a subscription of £5.00.
3. If possible a contribution such as, a request for information, article, software or philosophical thoughts.

S.J.U.G. contains the full spectrum of users from large installations in universities to small school systems with press-ganged teachers.

**S.J.U.G.**  
**Microtechnology Centre**  
**Wellington College**  
**Crowthorne**  
**Berkshire**  
**RG11 7PU**

**Contacts:** Humphrey Berridge  
Jan Edwards

**Interspan:** Humphrey Berridge of SJUG  
**Phone:** 0344-779020  
**Prestel:** 4017126953

Articles or letters can be sent to us directly, preferably as ASCII files or Wordwise files on disc. You can also send information or data to us via our Interspan mailbox.

# S.J.U.G. APPLICATION FORM

Please use block capitals throughout

Contact Name:
Address:
Post Code:
Tel. No.:

Type of SJ File Servers in use		
HDFS	Capacity:	
MDFS	Capacity:	Tape:
FDFS	Capacity:	

Number of Stations:
Number of Networks:
Length of Networks:

**I would like to become a member of the S.J. Research Users' Group. I enclose a cheque for £5.00**  
(Cheques should be made payable to "S.J.U.G.")

Signed:.....

Dated:.....

## S.J.U.G.

### S.J. RESEARCH USERS' GROUP

#### An invitation to join...

There are now quite a number of S.J. Research file-servers (and other Econet products) sold, and we have therefore formed a user group for those running these systems.

The S.J.U.G. allows users of S.J. products to exchange information (and possibly software), and to be in closer contact with S.J. themselves. There is a considerable amount of duplicated effort in a) writing network utilities, b) finding bugs, c) adapting various sorts of software to run on the networks. We see S.J.U.G. as being able to send out articles written by various users, letters requesting help or information, reviews of what software will run and what will not, and imaginative new ideas about Econet to exchange.

Do you have any ideas about what you would like to know, or things that you could write about for the benefit of other users? We have quickly produced the list below, but we are sure you can think of additions!

- # Fast file-handling on Econet.
- # Word processing on the net.
- # How do you handle graphics dumps?
- # Sideways ram.
- # Sideways ram utilities.
- # Problems with adapting disc software.
- # Licencing of network software.
- # Errors in the S.J. manual?
- # Errors in the Econet manual?
- # What utilities need adding?
- # Information on level III extensions.
- # What do you do with monitor output?
- # Local viewdata.
- # How do you incorporate Ceefax/Prestel?
- # Communicating with network via a modem.
- # Network mailbox.
- # Adding \*COMMANDS.
- # Music on the net.
- # What Econet utilities are worth buying?
- # What hardware faults have you had?
- # Lightening strikes and interfaces?
- # Differences between NFS3.34 and 3.60.
- # How to make the best use of printers?
- # Any security problems?
- # Multiple file copying to/from disc.
- # Bridges between networks.
- # Swapping/selling our own net software.

It must be stressed that S.J.U.G. is run quite independently of S.J. Research itself, but obviously we are in close contact with them for information and comments. We do not intend to make a profit from running the S.J.U.G. and will give our time in getting the group running - however we must cover our costs in printing, postage, phone, etc.

What we therefore ask is that if you would like to join S.J.U.G. - and we very much hope that you will -

a) Return to us the enclosed form - we will then put you on the mailing list, and send you copies of the Newsletter as they are published. We plan to publish three issues per year initially.

b) Enclose with the form a subscription of £2.50.

c) If at all possible, contribute something such as an article, software, request for information on xyz, or philosophic thoughts!

S.J.U.G. contains a wide spectrum of members, from those running large installations in university departments to small systems in schools being run by press-ganged teachers!

How to contact us:

S.J.U.G.  
Computing Dept.  
Wellington College  
Crowthorne  
Berkshire  
RG11 7PU

Contact:

Humphrey Berridge, or  
Alistair Shimmin

Phone: 0344-772137

PRESTEL: 4017126953

Articles or letters can be sent directly to us, preferably as ASCII or WORDWISE files (especially if they are long or contain listings!) on cassette, or 40 or 80 track disc. Later it may be possible to leave us messages via a modem and "mailbox" type of arrangement.

March 1986

H.J.J.B./A.R.B.S.

\*\*\*\*\*

PLEASE RETURN THIS FORM AS SOON AS POSSIBLE TO:

S.J.U.G., COMPUTING DEPT., WELLINGTON COLLEGE, CROWTHORNE, BERKSHIRE. RG11 7PU

I would like to become a member of the S.J. Research Users' Group. I enclose a cheque for £2.50.

(Cheques can be made payable to "S.J.U.G.")

Please fill in below (BLOCK CAPS.) so that we can check our records:

Contact name: (Mr/Mrs/Miss).....

Address:

.....  
.....  
.....  
.....  
.....  
.....  
.....

Phone: .....

S.J. Fileserver(s) used:

a) Hard disc: 20MB/40MB

b) Floppy: Capacity?

Number of stations running? .....

Signed: .....

Date: .....

\*\*\*\*\*

# NETWORK USER

LTS Publications,  
Haydon House, Alcester Road,  
Studley, Warks., B80 7AP  
Tel 0386 792617

The Computer Networks and Communications Magazine

## SPECIAL OFFER TO SJ FILE SERVER PURCHASERS

Congratulations, you are now the owner of what we consider to be one of the finest network management systems available. No doubt you will be concerned to ensure that you make the most of your investment.

If you are not already a subscriber to 'Network User' you may not know of the service to Econet users which we offer.

'Network User' is an independent bi-monthly magazine designed to assist educational users of network systems in selecting software and in deriving maximum benefit from the equipment they possess.

'Network User' carries regular reviews and features on software, telecommunications services, programming and applications across the curriculum. We also negotiate for our readers special discounts on software. Recent offers have included network adventure games, electronic mail and LOGO.

As a special offer to SJ File Server owners, if you take out a subscription using this form we will send you details of suppliers of educational software for Econet and the titles they publish, PLUS a network compatible disc containing the Network User Viewdata Terminal free of charge. The terminal is fully compatible with DTI modems and most others and is ideal for accessing viewdata services such as Prestel and the many free systems around the country. It includes facilities for saving and printing frames and for downloading Telesoftware.

An annual subscription to 'Network User' costs only £11.50 (6 issues) which we are sure you will agree is excellent value for money.

Please arrange an annual subscription to NETWORK USER at the rate of £11.50 (UK rate) £21.50 (Overseas rate) for 6 issues. (NB Cheques should be made payable to NETWORK USER)

- ☐ I enclose a cheque for £ .....
- ☐ Please invoice my School/LEA £ .....  
Order Number .....
- ☐ Please charge my credit card £ .....

Name .....

Organisation .....

Address .....

Card Name ..... Expiry Date ...../...../.....  
Number .....

We accept Barclaycard, Access.

Signature .....

Date .....

Post Code ..... Tel No.: .....

Ref SJNU

### PLEASE TICK WHERE APPLICABLE

Type of establishment

- ☐ Primary School
- ☐ Secondary School
- ☐ Independent School
- ☐ FE College
- ☐ Polytechnic
- ☐ University
- ☐ Teachers' Centre
- ☐ Other (please specify)

Size of Establishment .....

Type and No. of computers

..... No.:

..... No.:

Networks in use

..... No.:

..... No.:

Modems in use

..... No.:

..... No.:

Do you use Prestel ☐ TTNS ☐ ?

Job Title

- ☐ Head Teacher/Principal
- ☐ Deputy/Assistant Head
- ☐ Head of Department
- ☐ Senior Lecturer
- ☐ Teacher
- ☐ Lecturer
- ☐ LEA Centre Worker
- ☐ Adviser/Inspector
- ☐ Librarian
- ☐ Other (please specify)

This information will only be used to help in planning the magazine's content.

LTS Learning and Training Systems Ltd. Reg No 1773432 Directors M R Jones A W Jones M Sc. B Sc.

---

**Syntax:** \*COPY <source file specifier> <dest. file specifier>

## Description:

This program makes a copy of a file. Any user program or text in the computer is not corrupted as locations &900-&AFF are used as the buffer for transferring the data.

## Examples:

```
*COPY letter &.text.letter10
```

## Likely Errors:

Errors produced by \*LOAD, \*SAVE, OPENIN and OPENOUT can occur when running this program.

There are no other errors specific to this program.

## Associated Keywords:

COPIER, MULTICOPY

## Compatibility Notes:

Supported by Acorn systems.

## RISC OS Notes:

RISC OS command.

---

**Syntax: \*CV****Description:**

This program displays the station numbers of the currently selected File Server and Printer Server, the user's own station number, and the user id used to log on to the File Server. If there are multiple networks joined by bridges, then the network number will be returned. If the Printer Server is an SJ Research File Server then the name of the currently selected printer will be displayed next to the Printer Server station number.

If your computer has an old NFS rom such as NFS3.34 then a warning to this effect will also be displayed. Old NFS roms have some serious bugs and should be replaced with NFS3.60 or DNFS. An image of NFS 3.60 is supplied in the \$.LIBRARY on the release disc and you are free to erase the old NFS rom and replace it with the NFS3.60 image.

**Examples:**

\*CV

```
FS number 254
PS number 254 LASER
You are 001.005
User Id FRED
```

**Likely Errors:**

The message **RxCB ?** will be displayed if no receive control block is available in the BBC Microcomputer.

There are no other errors specific to this program.

**Associated Keywords:**

\*FSLIST, \*PSLIST

**Compatibility Notes:**

Supported by Acorn systems.

**RISC OS Notes:**

RISC OS version supplied in \$.ArthurLib.

**Syntax:** \*DESCRIBE [<topic>]

## Description:

This program displays help on various system utilities and also on error messages. The help is hierarchical so \*DESCRIBEing a topic may reveal further sub-topics on which help is available.

The program works by \*TYPEing text files in a !help sub-directory of the Library. It is possible to add your own help files to the system by preparing the text in a word processor and then spooling it out into the appropriate place in the !help directory structure.

## Examples:

```
*DESCRIBE
  Errors
  Utilities
```

```
*DESCRIBE Utilities.logon
```

```
Syntax
=====
*LOGON
```

```
Description
=====
```

A secure method of logging on to the file server as your password is not displayed.

```
Examples
=====
```

```
*LOGON
```

```
User Id :ARBS
Password:*****
```

## Likely Errors:

Errors produced by \*LOAD, \*SAVE and OPENIN can occur when running this program.

There are no other errors specific to this program.

## Associated Keywords:

```
*HELP
```

## Compatibility Notes:

Supported by Acorn systems.

## RISC OS Notes:

RISC OS version supplied in \$.ArthurLib. The RISC OS version access the same text files as the BBC version ie files in \$.Library.!help on drive 0.

# **\*FLUSH**

File Server command, controlling the built-in printer server

---

**Syntax:** **\*FLUSH** [<job name>]

## **Action with Wild Cards in Job Name:**

Occurs on every match

## **Description:**

This command causes printout to be flushed. It will be found useful if a user's program has generated large quantities of spurious output.

When a user issues this command without a parameter all printout sent by the user from that station will be cleared. This could include the job the file server is currently outputting to the printer.

To selectively remove files from the print queue **\*FLUSH** should be used with a job name. This name can include wildcards for deleting more than one file at a time.

Note that printers themselves often have an internal buffer, which means that they could carry on printing for some pages after a **\*FLUSH** command. To clear a printer's internal buffer, it will be necessary to turn the printer off and on.

## **Likely Errors:**

There are no errors specific to this command

## **Associated Keywords:**

**\*MFLUSH**, **\*PGO**, **\*PSTOP**

## **Compatibility Notes:**

Not supported by Acorn systems.

## **RISC OS Notes:**

Compatible with RISC OS.

## **Syntax: \*GOOODBYE**

### **Description:**

This command logs a user off all File Servers on a network. Open print jobs will also be closed. A message confirming successful logoff is displayed for each File Server.

### **Examples:**

```
*GOODBYE
      251 Logged off OK
065.019 Logged off OK
```

### **Likely Errors:**

There are no errors specific to this command.

### **Compatibility Notes:**

Supported by Acorn systems however Acorn File Servers will be displayed in the list of the File Servers logged of from several times, even if you were never logged on in the first place.

### **RISC OS Notes:**

Use the RISC OS \*SHUTDOWN command instead.

**Syntax:** \*MFLUSH <User Id> [<station number>]

## Description:

This command can flush multiple print jobs. It allows owners of the print queue directory to quickly remove large numbers of print jobs. The print queue directory is scanned for jobs which were printed by the specified user and these are then flushed using the \*FLUSH command. If the optional station number is also specified then only those jobs which were printed by the user at that station will be flushed.

When a print job is flushed the full info, as displayed by \*INFO, is output on the screen.

## Examples:

```
*MFLUSH BOOT
AA67      BOOT      at Stn.132 000123 /spl      hold      today 12:19 03F (000)
AB20      BOOT      at Stn.167 0004FD /spl      parall    today 13:45 03F (000)
BA01      BOOT      at 034.100 001E34 /spl      hold      today 16:01 03F (000)
```

```
*MFLUSH BOOT 132
AA67      BOOT      at Stn.132 000123 /spl      hold      today 12:19 03F (000)
```

## Likely Errors:

There are no errors specific to this command

## Associated Keywords:

\*FLUSH, \*PGO, \*PSTOP

## Compatibility Notes:

Not supported by Acorn systems.

## RISC OS Notes:

RISC OS version supplied in \$.ArthurLib.

## Syntax: CHAIN "MULTICOPY"

### Action with Wild Cards in the Directory Name:

Occurs on first match (alphabetically).

### Description:

This program copies entire directory trees between File Servers, or between different places in the same File Server.

It will prompt for the log-on text for the File Server containing the source files, and the same for the destination File Server.

The program will then ask **Do you wish to force overwriting of locked files (Y/N)**: -- if the user answers Y, it will overwrite existing files even if they are locked. The next questions asks **Do you wish to include sub-directories (Y/N)** -- if the user answers Y, it will copy the entire set of sub-directories and the files in them. It will also ask if the account information is to be copied -- if the answer to this question is N, then all files and directories will be put in the main account of the destination directory.

There is also an option to copy the creation dates of the files; this is intended for use when backing up the File Server. The system manager may set this option so that ordinary users cannot use it.

The user must own the destination directory, and have read access to all the files to be copied.

### Examples:

```
CHAIN "MULTICOPY"  
Multiple file copy utility V1.11
```

MULTICOPY copies groups of files from one file server to another. It may also be used between directories or discs on the same file server.

```
Log-on text for source FS  
(or press RETURN to use current FS) :  
*I AM 254 FRED  
Log-on text for dest. FS  
(or press RETURN for same FS) :  
*I AM 253 FRED  
Do you wish to force overwriting of  
locked files (Y/N) :Y  
Do you wish to include sub directories (Y/N) :Y  
Do you wish to copy account information (Y/N) :N  
Do you wish to copy creation date etc.  
(for system manager's use only) (Y/N) :Y  
source directory name :PROGS  
destination directory name :PROGS  
.  
.  
.
```

(list of the files being copied)

The next example shows a copy of the directory structure '\$.RELEASE' from a hard disc called 'MAIN1' to a floppy disc called 'Main2', on the same File Server.

CHAIN "MULTICOPY"

Multiple file copy utility V1.11

MULTICOPY copies groups of files from one file server to another. It may also be used between directories or discs on the same file server.

Log-on text for source FS

(or press RETURN to use current FS) :

\*I AM **254 FRED**

Log-on text for dest. FS

(or press RETURN for same FS) :

\*I AM

Do you wish to force overwriting of

locked files (Y/N) :**Y**

Do you wish to include sub directories (Y/N) :**Y**

Do you wish to copy account information (Y/N) :**N**

Do you wish to copy creation date etc.

(for system manager's use only) (Y/N) :**N**

source directory name :**\$\*1.RELEASE**

destination directory name :**\$\*2.NEWREL.ANOTHER**

·  
·  
·

(list of the files being copied)

## Error Handling:

If an error occurs during a copying operation then MULTICOPY will produce a \* prompt at which you can type the commands necessary to fix the problem. In particular it allows errors such as **Already open by**, **Account bankrupt** and **Too short** to be fixed. When all the necessary \* commands have been typed pressing just <RETURN> at the \* prompt will bring up the prompt **R(etry) or S(kip)**. Selecting **R** means MULTICOPY will resume trying to copy the file which caused the error. Selecting **S** means that the file which caused the error will be skipped and copying will resume at the next file. This is important should you encounter a file which cannot be copied due to an error which cannot be fixed, eg a disc error.

## Backing Up to MDFS Floppy:

MULTICOPY can be used to backup to MDFS format floppy discs. This is achieved by simply entering the source and destination directories with the relevant disc name prefix. With care it is possible to swap floppy discs during the copying operation and thus backup more than 800k in one operation. Log on to the File Server using a user name from the password file on the hard disc. Create a set of floppy discs for backup purposes and name them Backup1, Backup2, Backup3 etc. As the source directory specify the hard disc eg **\$HARD** and as the destination directory specify the floppy disc using a wildcard which will match all the floppy discs names eg **\$Backup\***. Eventually the copying operation will produce the **Disc full** error and hence the \* prompt. At this moment the program is still accessing the floppy disc so removing the disc would cause a fatal error. Press <RETURN> to bring up the **R(etry) or S(kip)** prompt. At this moment the program is accessing the hard disc so it is possible to press the **Release Discs** button and swap to the next backup floppy. Press **R** to retry the file which caused the previous floppy disc to become full and copying will be resumed. All the necessary sub-directories will be created on the new floppy disc.

## Likely Errors:

### Insufficient access

#### Error 189 (BD)

If the user does not have access **R** to all the files to be copied from the source, or does not own the destination directory.

### xxxx is not a directory

#### Error 190 (BE)

If the user has specified a file as the source or destination directory name prompt.

### Already opened by xxxx

#### Error 195 (C2)

MULTICOPY will save over a file of the same name. If this file was already open, this error will occur.

### Locked

#### Error 195 (C3)

After an attempt to save over a file of the same name, if the latter was locked.

### xxxx Not Found

#### Error 214 (D6)

If the source directory could not be found.

### Account xxxx bankrupt

#### Error 198 (C6)

If the account number being saved to does not have sufficient credit.

## Associated Keywords:

COPIER

## Compatibility Notes:

Supported by Acorn systems, except that accounts do not exist, and so an attempt to copy account information across will cause an error. Since Acorn systems use the root of user's tree of directories to determine its ownership (rather than account numbers), a user will not have owner access to files specified as `$.{<directory name>}<file name>`. To get round this problem, it is wise to log on as a system privileged user.

## RISC OS Notes:

Use either the RISC OS `*COPY` command or the desktop copy facility instead.

---

## Syntax: \*NEWPASS

### Description:

This command allows a user to change their password. It is the most secure method as it makes use of an encryption key to encode both the user's old password and the new one, before sending the change password command to the File Server.

The new and old passwords are not displayed on screen but keypresses are echoed as asterisks. The New Password prompt will appear twice and the user must enter the same new password twice. This is to prevent the user from accidentally setting their password to something they do not know as the result of a mistype. If the two entries of the new password are not the same the **Retype PW** error message will be given and the user will need to type \*NEWPASS again.

A minimum password length of six characters is recommended for the encryption algorithm to be fully effective.

### Examples:

```
*NEWPASS
Old Password: *****
New Password: *****
New Password: *****
```

### Likely Errors:

#### Password file changed                      Error 3 (03)

This error will be produced if the password file has been changed by the system manager, while the user is logged on. The user should log on again.

#### Bad password                                      Error 185 (BD)

There is an illegal character in the password quoted, probably \* # \$ % ^:

#### Wrong password                                  Error 187 (BB)

The old password does not match the one stored.

### Associated Keywords:

\*LOGON, \*PASS, \*I AM

### Compatibility Notes:

Not supported by Acorn systems.

### RISC OS Notes:

RISC OS version supplied in \$.ArthurLib.

## Syntax:\*PSLIST

### Description:

\*PSLIST displays a list of all of the available printer servers on the network. For SJ Research printer servers, the logical printers available on each File Server will be listed after the station number.

Logical printers will not be listed if they have been set by the system manager to be non-existent. However, all other logical printers on an available printer server will be listed. Those you are not allowed to use will be prefixed with the x character.

If the Eiconet installation comprises multiple networks, \*PSLIST will also display printer servers on other networks, preceded by their network number (e.g. printer server 235 on network 2 will be displayed as 002.235).

The status of the physical printer appropriate to the currently selected logical printer will be given by \*PSLIST after the printer server station number. The status codes currently supported are:

**Ready** -- this printer is ready to start printing, or that print spooling is available for this printer.

**Busy with nnn** -- this non-spooling printer is busy printing output from station nnn.

**Jammed** -- this printer has jammed with paper, has run out of ribbon or some similar event. For a print-spooling printer, the directory %PRINTQ may be full or not found. Jammed printers will not accept any data.

If you are not allowed to use the logical printer which is currently selected for you on a printer server, \*PSLIST will not list that station. Note that it is not possible to select a logical printer you cannot use with the commands \*PS and \*PRINTER; so there are only three reasons why a printer server should not be listed:

1. You may not be allowed to use any logical printers on that printer server. The logical printers may not exist, or they may require users to be logged on the File Server or to have access to a particular account.
2. You may not have changed your logical printer on that printer server since you logged on, and the default printer on that File Server may not be available to you.
3. You may have selected a logical printer on that printer server with \*PS or \*PRINTER when you had access to it; but the system manager has edited the printer information so that you are no longer allowed to use that logical printer.

### Examples:

```
*PSLIST
000.235 Ready
      EPSON      Mac      x NOBANN      LASER      CONDEN      PSCRIPT
NONSP      MCMULT000.251 Ready
      parall      serial      hold      auto
200 Busy with 023
```

**Likely Errors:**

There are no errors specific to this command.

**Associated keywords:**

\*PRINTER, \*PS

**Compatibility Notes:**

Supported by Acorn systems, except that Acorn systems do not support logical printers.

**RISC OS Notes:**

Not available on RISC OS.

**Syntax:** CHAIN"ACCLIST"

## Action with Wild Cards:

Not applicable

## Description:

This program gives information on the use of accounts in the %PASSWORD file. This is useful for checking that account numbers have not been doubly allocated and also for finding which users own a particular account number.

If the password file is large then it can take some time for the program to build the map of account numbers used. While this process is taking place dots are printed on screen, one for every ten users in the password file.

## Example:

CHAIN"ACCLIST"

Scanning password files.....

- A. Summary of personal accounts
- B. Full details of a single account
- C. Users with no personal account
- D. Summary of group accounts
- E. Block accounts
- F. Do all the above
- G. Exit

Enter choice :

**A** will list all account numbers allocated as a personal account. If two users share a personal account then both their User Ids will be displayed. If more than two users share a personal account then the first two names in the password file will be displayed plus the number of other users who own the account.

**B** will list all the users who own the specified account.

**C** will list all users who have not been allocated a personal account.

**D** will list the first user owns the account number plus the number of other users who own the account.

**E** will list owners of the blocks of sixty-four accounts above & 100.

**F** performs all of the operations A-E in sequence.

## **Likely Errors:**

### **Key Locked**

### **Error 5 (05)**

If the key switch on the front panel of the MDFS is not in the SYST position.

### **Insufficient privilege**

### **Error 186 (BA)**

If the user does not have system privilege.

## **Compatibility Notes:**

Not supported by Acorn systems.

## **RISC OS Notes:**

ARCACCLIST is the RISC OS version.

## **Syntax : \*BACKUP [C]**

### **Description :**

This program displays details of any tape backup currently pending. The optional C parameter can be used to cancel a pending backup.

### **Examples :**

#### **\*BACKUP**

Disc : HardDisc1  
Tape : Archive#1  
Backup at: 23:00 on 23/04/89  
Printer output: Parallel

#### **\*BACKUP C**

Disc : HardDisc1  
Tape : Archive#1  
Backup at: 23:00 on 23/04/89  
Printer output: Parallel

Cancel (Y/N) Y

### **Likely Errors:**

#### **Key Locked**

#### **Error 5 (05)**

If an attempt is made to cancel a backup and the key switch on the front panel of the MDFS is not in the SYST position.

#### **Insufficient privilege**

#### **Error 186 (BA)**

If the user attempts to cancel a backup and does not have system privilege.

### **Compatibility Notes:**

Not supported by Acorn systems.

### **RISC OS Notes:**

RISC OS version supplied in \$.ArthurLib.

---

**Syntax:** \*DESTROY <general specifier> <disc name>

## Action with Wild Cards:

Wild cards prohibited.

## Description:

The \*DESTROY command is used primarily for recovering from **Disc Read Only** errors caused by a corrupt directory structure (accompanied by the messages **Wrong number of files in dir.** or **Bad Backpointer**). It is only supported by File Server versions 0.A4 or later.

Its effect is to destroy the directory (and any files or subdirectories therein) by turning it into a file. A subsequent reboot will ignore the bad area of the directory structure and allow writing to the disc again. You must have system privilege and owner access to the directory in order to use the command.

N.B. this command has the capability to destroy the root, \$, with obviously dire consequences. You have been warned.

<directory name> may contain wild cards, but only the first match will be acted upon. This is useful in getting rid of a file or directory with bad characters in it, but only with caution as you must be sure that the wildcard specifier will not accidentally match with the wrong file.

<discname> is the name of the disc on which the directory is being destroyed. It must be specified in full, i.e. with no wild cards, this being a built-in safety feature.

## Example:

Wrong number of files in dir. on drive 0  
in WORK  
in PAUL  
in \$

has appeared on the printer after a reboot. The fileservr should be on line, but whenever any writing is attempted to that disc the error message **Disc Read Only** will be given. Suppose that the disc 0 is called **PUPIL-DISC**.

The most important thing to do now is to logon, select the directory (\$.PAUL.WORK) and catalogue the directory. It may be possible to recover some of the files in this directory, and if you wish to do so you must take the opportunity now. (It may be that you have a recent backup of these files on a tape or other medium and so you need not bother). They can be copied onto another disc on the fileservr or onto local floppy discs.

Now logon as a system privileged user, and turn the key to the System position. Select the correct disc, and type the command

**\*DESTROY \$.PAUL.WORK <SPACE> PUPIL-DISC<return>**

You will now discover that \$.PAUL.WORK is a file. Reboot the fileservr (by pressing the Release Discs button twice on an MDFS, or by powering-off/on with an HDFS). You can now delete the file \$.PAUL.WORK because the disc is now read/write. (\*DESTROY is the only command that will write to 'read only' discs, and this includes physically write-protected floppy discs).

The message **Bad Backpointer in <dir>** is slightly more difficult to recover as the directory specified is not corrupt; one of its subdirectories has a bad backpointer. To find out which one is corrupt, select the specified directory and take a catalogue. For each of the subdirectories listed, type **\*INFO**

<directory name>.^ until the name given is different to your current directory. You have now identified which directory has the bad backpointer. This directory will have to be destroyed, so take a copy of all the files in that directory (all of which will be recoverable) and then use the \*DESTROY command as above.

You can also use \*DESTROY to remove an overly-large or overly-deep directory structure. But remember that \*DESTROY is a very dangerous command and should only be used when absolutely necessary (given that the more you use it, the more likely you will make a mistake). To be absolutely safe, take a backup of the disc before using the command.

## **Likely Errors:**

### **Key Locked**

### **Error 5 (05)**

If the key switch on the front panel of the MDF5 is not in the SYST position.

### **Insufficient privilege**

### **Error 186 (BA)**

If the user does not have system privilege.

## **Compatibility Notes:**

Not supported by Acorn systems.

## **RISC OS Notes:**

Compatible with RISC OS.

**Syntax:** CHAIN "EDITPASS"

## Description:

The system maintains a special file %PASSWORDS, which does not exist in any particular directory, and is only 'visible' to system privileged users. The password file cannot be deleted with \*DELETE, but can only be cleared in Utility Mode, although it is possible to save over the top of it. The name must be quoted in full in upper case, i.e. no wildcards. EDITPASS is provided in the library for editing the contents of %PASSWORDS.

EDITPASS is a screen editor for editing password files, or for creating new ones. The present version can handle about 200 users, and requires a BBC computer with 32K RAM (i.e. Model B or expanded Model A). If a BBC with a 6502 second processor is used, more than 300 users can be created.

(There is a version of this program, **EDITPASS!**, which uses more readable identifiers, if the system manager wants to see how the program works or to make his own version. EDITPASS itself has been condensed, so as to leave maximum storage for data).

Although anyone can run EDITPASS (assuming that the system manager has set public read access to this file), it is necessary to have system privilege *and access to account 0*, and the MDFS front-panel keyswitch must be in the "SYST" position to either read or save the password file on a disc.

EDITPASS always works with the currently selected disc drive: to edit the password file on another drive, use \*DIR :<disc name> to select this new disc.

*Be cautious when running this program. All the system passwords will be loaded into memory, and may be displayed on the screen. Never walk away from the computer running EDITPASS without typing \*BYE and switching off the power.*

When the program is run, the program will prompt:

```
Password file Editor 27feb86
```

```
Maximum number of users=257
```

```
Options:
```

```
  E - edit PW file from disc
```

```
  O - edit file from current RAM
```

```
?
```

The normal option is E, unless it has been necessary to delete the password file for some reason. The O option is useful if this program has stopped, either with an error, or as a result of pressing the <Escape> or Q key.

The program will then display all the users, with their boot options and system privilege (if any). The display will be similar to:

Add	Delete	Find	Save	eXpand	Quit	*
User id		Boot option		Privilege		
ARG		0 Off				
BASHER		0 Off	System			
BRIAN		0 Off				
BOOT		3 Exec				
CLAIRE		0 Off				
DEFAULT		2 Run	*DEFAULT*			
JEF		0 Off				
JOHN		3 Exec				
KIM		2 Run				

The 'up' and 'down' cursor keys scroll the display, allowing the total user list to be available. To add new users, press the **A** key: the program will prompt repeatedly for names until the list is terminated with <Return> on its own.

**D** key will delete the user on the current line.

**\*** key allows normal **\*** commands to be run from within the program. To return to the list of users, type <Return> on its own on a line.

**Q** key stops the program without saving the result back to disc. The <Escape> key, pressed at any stage, has the same effect, except in the **\*** mode.

**S** key saves the password file to disc, and stops. A check will be made that there is at least one system privileged user, and that a user exists with access to account 0 (these are both vital to the running of the system), and the program will print a warning if one of these requirements is not met. Note that you may however want to keep all your system users on one (or a few) discs for security, in which case it would be legitimate for there to be no system user, or user with access to account 0, on other discs. The name of the *default user* if any (see below) will be displayed, and then the prompt Save (Y/N) : Typing N will return to the list of users.

After typing **Q** or **S**, at the end of the system manager's session, type **\*BYE** then **TURN OFF THE BBC MICROCOMPUTER** at which you were logged on. The password file will remain in the computer unless it has been overwritten, and another user could easily read it from there, and gain unauthorised access to the system.

**X** key enters the *expanded mode* for the user at the current cursor position. The display will change to, for example:

```

A(ccount)  S(ystem priv) D(efault user)
P(assword) U(RD) L(IB) 0..3 Boot option
G(roup Ac) E(nable) R(un only)
6-AuxLock 7-No Lib 8-Saves 9-PW lock

FRED              0 Off
PW : CRICKET
URD: USR.ARG      Short SAVES OK
LIB: (normal)     Fixed *ENABLE

```

Personal account number : 1D8

Accounts : 0->15 25 60->6F F0->FF

Expanded mode allows the details of each user to be edited. New users are initially created with no password set, boot option 0 (see under **\*OPT4**, Section 6.6), normal library and user root directories, and access to no accounts at all.

- 0 1 2 or 3 will set the boot option to that value. The boot option may also be set by the users themselves, unless the PW lock option has been set (see below).
- 6 will turn on and off an option to prevent users changing the auxiliary account number of a file or directory. This prevents problems with users experimenting with auxiliary account numbers and consequently losing access to their own files.
- 7 will turn on and off an option to give a user an Acorn style library search. The full SJ library search will occur on any load or open command eg OPENIN, CHAIN, \*LOAD. An Acorn style library is only searched when \*RUN commands are issued.
- 8 will turn on and off an option to prevent users from saving a file shorter than 16 bytes in length with the SAVE or \*SAVE command. This option helps to avoid the problem where BASIC (for example) will save a null file if an attempt is made to save after pressing <Break> without typing "OLD".
- 9 will turn on and off an option to prevent users from changing their password and boot option. It could be useful to set this option for the default user, to prevent unauthorised users from changing the default password and option.
- S toggles system privilege off and on for that user. *Note that there must be at least one system privileged user on the system, or it will not be possible to change the password file thereafter.*
- D sets this user as the default user. There can be only one default user, so this command will change the default to this user. Keying D again will remove the default setting, so that there is no default user. Users logging-on to the File Server with unrecognised user identifiers will be logged on as the default user -- the system manager should set up a boot file to re-direct them, if necessary. If there is no default user, the error User not known will be displayed.
- E toggles an option requiring the user to type \*ENABLE before attempting \*DELETE with a wild-card specifier, as a safety measure.
- P prompts for a password. Users can also set up their own passwords with the \*PASS command unless the PW lock option has been set. Passwords can be up to 10 characters long, and have the same valid characters as file names.
- A prompts for the user's *personal* account number. This should be unique, and can be any number in the range 1 to 7FF. Account number zero is reserved for the system's use; if this is typed, then no personal account will be allocated.
- G prompts for *group* account numbers. In response, it is possible to type a single account, a list of accounts separated by commas or spaces, or a range of accounts: for example 1-55 specifies all accounts from 1 to 55. Adding a minus sign to the start of the line causes the specified accounts to be removed from the list, rather than being added. Typing just <Return> will leave the shared accounts the same. Any *shared* account numbers above FF will be allocated in blocks of 64 accounts (40 hexadecimal). That is entering '1AB' will result in the block of accounts from 180 to 1BF being added. Note that account 0 should normally be allocated only to the system manager.
- U prompts for a user root directory. This can be a path name going through several directories, and can be up to 80 characters long. Disc names can also be included to specify a disc; the default disc is the one in which the user is found in the password file. If <Return> is pressed, the <-normal-> option of a directory with the same name as the user identifier will be selected. If the specified URD is not found on logging-on, the user will be in the root directory of the default disc, even if another disc was specified. Wild cards can be used in a URD specification, although this is not recommended.

L prompts for a default library directory for the user. This can be a path name going through several directories, and can be up to 80 characters long. If <Return> is pressed, the <-normal-> option of \$.LIBRARY on the lowest numbered disc drive is selected. If the specified library is not found, the default library will be set to the root directory on the lowest numbered disc. Wild cards can be used in a library specification, although this is not recommended. The character & can be used as a synonym for the matching URD if required.

To return to the list of users, press <Return>.

### **Likely Errors:**

**Key Locked**

**Error 5 (05)**

If the key switch on the front panel of the MDFS is not in the SYST position.

**Insufficient privilege**

**Error 186 (BA)**

If the user does not have system privilege.

### **Compatibility Notes:**

Not supported by Acorn systems.

### **RISC OS Notes:**

ARCPASS is the RISC OS version.

**Syntax:** CHAIN "EDITPRINT"

## Description:

The EDITPRINT program allows the system manager to set up details for the two printers which can be connected to the File Server.

Although anyone can run EDITPRINT (assuming that the system manager has set public read access to this file) to find the default settings, it is necessary to have system privilege, and the MDFS front-panel keyswitch must be in the "SYST" position to change the printer information.

This description merely describes how the program works : see Chapter 6 for advice on suitable values to set.

To adjust the printer settings, type:

**CHAIN "EDITPRINT"**

The program will respond with the editing screen similar to the one below :

- EDITPRINT - Printer Setup for SJ Research File Server 254						
Name	Status	Output to	Anonymous Printing	Account Required	AccNo	Default
parall	Spooling	Parallel	No	No		
serial	Spooling	Serial	No	No		
hold	Hold		No	No		
auto	Auto	1st:parall 2nd:serial				
	Off					
	Off					
	Off					
	Off					
	Off					
	Off					
	Off					
	Off					
	Off					
	Off					
	Off					
	Off					
	Off					
	Off					

Esc: Quit

Space: Toggle Data

f3: Save Details & Exit

You can move round the screen using the cursor keys. Your current position in the screen is shown by displaying the field in reverse video. Editing a field can be done by pressing <Space> to rotate through the possible legal values. Alternatively pressing the initial letter of the value you want will enter that value.

Although there are only two physical printers attached to the File Server there are 16 logical printer names which can be used to access these printers. Each of these logical printers can have a different set of properties and these are displayed in the row across from the name.

Name is the name which users will quote to specify that particular logical printer. Printer names may be up to six characters long. The name PRINT is reserved and must not be given to a printer. If the printer name is blank (i.e. consists of spaces), that printer is disabled completely.

**Status** determines where output sent to this logical printer will be directed. There are four possible values.

- |              |   |
|--------------|---|
| Off          | This disables the printer completely so that it cannot be selected using either *PS or *PRINTER   |
| Spooling     | Printer output sent to a spooling printer is stored as a print job file in the %PRINTQ directory prior to being sent to the printer.  |
| Non-spooling | Printer output sent to a non-spooling printer will be output immediately if the relevant physical printer is free. If the physical printer is busy then output will be spooled.   |
| Hold         | Printer output sent to a hold printer will be stored as a print job file in %PRINTQ however the File Server will not attempt to output the data stored to a physical printer.   |
| Auto         | Printer output sent to an auto printer will go to one of two possible other logical printers, a first choice and a second choice. At the time of printing the File Server will attempt to allocate the first choice printer and if this is not possible will try to allocate the second choice printer. |

**Output to** determines which physical printer will be used for output sent to a logical printer. In the case of an auto printer this column will hold the first and second choice logical printers.

**Anonymous Printing** controls whether users who have selected this logical printer, but are not logged on to the File Server, may print.

**Account Required** controls whether a user requires a specific account number to select this logical printer.

**AccNo** specifies the account needed if Account Required has been set to Yes.

**Default** specifies whether this is a default printer or not. If a user attempts to print without explicitly selecting a printer by name then the File Server will try to allocate a default printer. The File Server works down the list of printers defined as default printers and will allocate the first default printer which the user is allowed to use.

Having edited details they are saved by pressing the  $f_3$  key. Before saving some checking is performed to ensure that in particular there is no combination of auto printers which is circular. If this is found to be the case then an error is given and the cursor is move to the printer definition which is incorrect.

## Likely Errors:

**Key Locked**                      **Error 5 (05)**  
If the key switch on the front panel of the MDFS is not in the SYST position.

**Insufficient privilege**              **Error 186 (BA)**  
If the user does not have system privilege.

## Compatibility Notes:

Not supported by Acorn systems.

## RISC OS Notes:

Compatible with RISC OS.

## Syntax: CHAIN"LISTUSERS"

## Action with Wild Cards:

Not applicable

## Description:

This program lists all the users, held in the %PASSWORDS files, with their accounts and any special options that are set. This is especially useful for providing the system manager with a list of information so that he knows which accounts are spare to allocate.

If your system has more than one password file then a list of these will be displayed. You can select which password file to list or simply pressing <Return> to produce an alphabetically merged list of all the users in all the password files.

To send the output to the printer type CTRL B before running the program.

## Example:

CHAIN"LISTUSERS"

Scanning for password files...

Drive 0 \$R01 OK password file

Enter drive number or <RETURN> for all :

Name=BOOT	Pacc=	Boot=3	Flags=Pw
Low accounts		High accounts	
URD=(normal)			
LIB=(normal)			
Name=FRED	Pacc=1D8	Boot=0	Flags=NsAl
Low accounts		High accounts	
URD=USR.DEMO.FRED			
LIB=(normal)			
Name=SYST	Pacc=	Boot=0	Flags=SyEn
Low accounts 0-FF		High accounts 100-7FF	
URD=(normal)			
LIB=(normal)			

## Likely Errors:

Key Locked Error 5 (05)

If the key switch on the front panel of the MDFS is not in the SYST position.

Insufficient privilege Error 186 (BA)

If the user does not have system privilege.

## Compatibility Notes:

Not supported by Acorn systems.

## RISC OS Notes:

ARCLSTUSRS is the RISC OS version.

**Syntax :** \*NEXTPACC <Account number>

**Description :**

This program displays the next unused personal account number in the block of sixty-four accounts containing the specified account. If no free account exists in this block of sixty-four then the search will continue in to higher numbered accounts.

**Examples:**

\*NEXTPACC 240  
0278

**Likely Errors:**

**Key Locked**

**Error 5 (05)**

If an attempt is made to cancel a backup and the key switch on the front panel of the MDFS is not in the SYST position.

**Insufficient privilege**

**Error 186 (BA)**

If the user attempts to cancel a backup and does not have system privilege.

**Compatibility Notes:**

Not supported by Acorn systems.

**RISC OS Notes:**

Not available for RISC OS.

**Syntax:** CHAIN "SETBACKUP"**Description:**

This program is used for setting the time of an auto backup.

When run the File Server will set a time at which it will go offline to perform a tape backup. This automatic backup can be cancelled using this program or with the \*BACKUP utility. Any pending backup is always discarded when the File Server is switched on, or when it is put into Utility Mode. If the tape intended for backup is removed from the File Server then a system message will be printed to the effect that the backup is no longer possible. This is to alert anyone swapping tapes to the fact that they will need to replace the tape if the backup is to proceed as planned. If the tape is not replaced a system message will appear once every hour until such time as the correct tape is replaced.

The screen display is mostly status information on the tape currently loaded and any backup currently pending.

```

                                - SETBACKUP -
                        Set Time for Backup on SJ Research File Server 254

```

---

```

Current File Server Time:  20:01 on 13 Apr 89
Schedule for Next Backup:  Disc: Work                      Tape: Archive#1
                           Time: 23:00 on 13 Apr 89      Output report: Parallel
Tape Status:
    Tape Name:             Archive#1
    Description:           This tape only for backup of 'Work'
    Number of Passes:      536 (10% of nominal life expectancy)
    Tape Formatted:        14:10 on 01 Sep 88
    Current contents:      Disc "Work" (Status: OK)
    Backed up at:          23:30 on 12 Mar 89

```

---

```

Disc name to back up:      Work
Time to start Backup:      23:00 on 13 Apr 89
Output report:             Parallel
Time to next backup:       0 days, 02 hrs, 59 mins

```

---

```

Esc: Quit      + - Space: Change Data      f3: Set Backup      f9: Cancel Backup

```

To change an entry simply highlight the relevant field and press <Space> to rotate through possible legal values. The date of the backup is changed using the + and - keys. The default disc name to be backed up will be the same as the name of the disc stored on the tape. If no disc name matches that on tape then the default will be the disc in drive 0. Should the tape be removed from the tape drive when a backup is pending then the File Server will print a system message warning that the pending backup will now fail. This message will continue to be printed at five minute intervals. When the correct tape is replaced in the tape drive a system message will be printed informing that the intended backup can now occur.

## **Likely Errors:**

### **Key Locked**

### **Error 5 (05)**

If the key switch on the front panel of the MDFS is not in the SYST position.

### **Insufficient privilege**

### **Error 186 (BA)**

If the user does not have system privilege.

## **Compatibility Notes:**

Not supported by Acorn systems.

## **RISC OS Notes:**

Compatible with RISC OS.

## Syntax: CHAIN "SETSYSTEM"

### Description:

This program is used for setting various system functions.

This includes the printer port which will be used for the output of system messages, the level of system message reporting and the privilege needed to set the time of the File Server clock.

- SETSYSTEM -  
System Setup for SJ Research File Server 254

Function	Status
Privilege needed to change File Server time	None
System messages printer port	Parallel
System message level	000 = Off

Key to System Message level options...

000 = Off	015 = *cat & opens
005 = Logon/Logoff	128 = Aborted loads
007 = Errors	130 = Fn codes
010 = Maximum users & * commands	150 = Net errors
011 = Load/save	

Esc: Quit

Space: Toggle Data

f3: Save Details & Exit

To change an entry simply highlight the relevant field and press <Space> to rotate through possible legal values. In the case of the System Message level it is possible to enter the level you want by typing it directly. The possible System Message levels are displayed in a help table on screen. These levels are cumulative (10 includes the messages produced by 7 and 5). The normal system message level is 0 however serious system errors will always be printed.

### Likely Errors:

**Key Locked**                      **Error 5 (05)**  
If the key switch on the front panel of the MDF5 is not in the SYST position.

**Insufficient privilege**              **Error 186 (BA)**  
If the user does not have system privilege.

### Compatibility Notes:

Not supported by Acorn systems.

### RISC OS Notes:

Compatible with RISC OS.

---

**Syntax:** CHAIN "SETTIME"

**Description:**

This program sets the internal real-time clock in the File Server .

The program displays the current File Server date and time, and allows this to be altered. The time is set at the moment the press the space bar to set the clock to the time entered, allowing accurate synchronisation with the speaking clock or Greenwich pips.

**Example:**

The program display will be similar to the one below:

```
          - SETTIME -  
Set clock on SJ Research File Server 254
```

---

```
Hrs:14  Mins:22  Secs:17          Day:14  Month:Apr  Year:1989
```

---

---

```
Esc: Quit          + - Space: Change Time          f3: Write Time & Exit
```

---

The cursor keys are used to moved between the different fields in the date and time. The + and - keys are used to alter the contents of a field either up or down.

**Likely Errors:**

**Key Locked**                      **Error 5 (05)**  
If the key switch on the front panel of the MDFS is not in the SYST position.

**Insufficient privilege**              **Error 186 (BA)**  
If the user does not have system privilege.

**Compatibility Notes:**

Supported by Acorn Level 3 systems.

**RISC OS Notes:**

Compatible with RISC OS.

**Syntax:** CHAIN "SYSADM"

**Description:**

This BASIC program is a front end menu to all the system management programs. The program resides in the library so that it always provides quick and easy access to system management programs. To run any of the management programs simply highlight the program required and press return.

## **Syntax: \*TAPEINFO**

## **Description:**

This command displays the tape id sector of the tape currently loaded in the tape drive. It is useful for checking the status of a tape which has been used for backup using the automatic backup facility

## **Example:**

### **\*TAPEINFO**

Tape name: G-Daily#1

Passes: 2048

Description:

Disc name: Work

Backed up at: 04:49 on 14/06/89

Status: OK

## **Likely Errors:**

Tape Cartridge not found      Error 214 (D6)

## **Compatibility Notes:**

Not supported by Acorn systems.

## **RISC OS Notes:**

RISC OS version supplied in \$.ArthurLib.

## 5.5 Password File Management System

### 5.5.1 Overview

#### Batch mode Editor Documentation:-

- 5.5.1.1 Memory Requirements
- 5.5.2 \*CONVERT
- 5.5.3 \*MERGE
- 5.5.3.1 !mkdir and !rmdir
- 5.5.4 \*GENERATE
- 5.5.5 Keywords
- 5.5.6 Mod-file Examples
- 5.5.7 Errors
- 5.5.8 Formal File Definition
- 5.5.9 Known Problems

### 5.5.1 Overview

The password file management system software consists of the following programs, all of which are found in the directory \$. SYSPROGS of the release disc:-

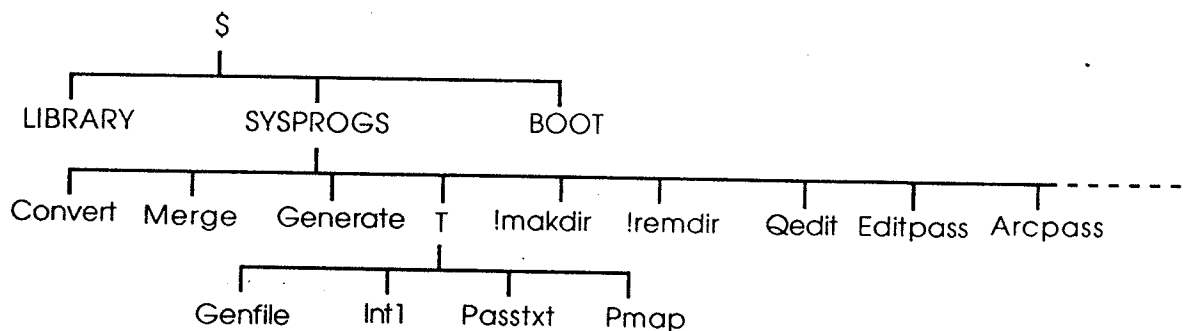
#### a) The batch mode editing suite:

CONVERT (Machine code program)  
MERGE (Machine code program)  
GENERATE (Machine code program)

#### b) The interactive editors:

QEDIT (BASIC program)  
EDITPASS (BASIC program)  
ARCPASS (Archimedes BASIC program)

The directory structure is shown thus:-



The existing EDITPASS program restricts the size of the password file to the size of the memory in the local computer, and this typically allows around 200 users. There is now a version of version of this program (called ARCPASS) for an Archimedes allowing about 7000 users. The batch mode editor and QEDIT are a means of editing large password files on standard BBC microcomputers.

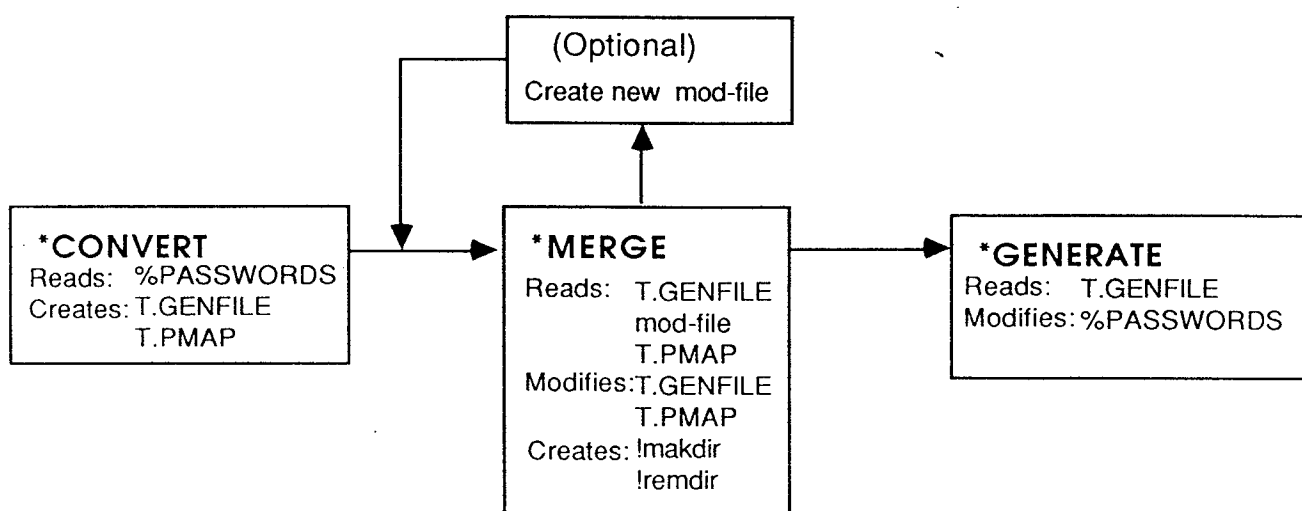
QEDIT is a version of EDITPASS which allows the password file to be edited on a user-by-user basis. The password file is not held in the local computer; individual user entries are modified and then written back to the password file directly, so the restriction on file size is removed. However, QEDIT does not allow you to insert or remove users, or to change the URD or LIB strings.

With the batch mode editor, the system manager prepares a text file (the *mod-file*) containing instructions for modifying the password file. The commands available can be very powerful; for instance, the system can automatically allocate a spare account number, create the appropriate user directory, set its account number

and credit that account. The same process can be repeated automatically, so with little more than a list of names, an entire class can be entered onto the system in a matter of minutes.

The batch mode editor uses a three-stage process: \***CONVERT** converts the (machine-readable) password file %**PASSWORDS** into a (human-readable) text file. \***MERGE** combines this with the mod-file to produce another text file. \***GENERATE** converts this text file back into a machine-readable format file, which it then installs on the appropriate disc.

The process is shown in the diagram below:-



### Important

Since all of the passwords are stored in the text files, it is very important that only the system manager has access to them, and they should be treated with as much respect as %**PASSWORDS** itself. Each of the programs protects the machine from remote network operations to stop unauthorised people being able to read the files, but security is only as good as the system manager makes it. The T directory should be set to Private (\***ACCESS** T +P). **Only %**PASSWORDS** is protected by the key: the other files are only protected by the main file access controls.**

### Off-line / Off-site Operation

An advantage of the batch mode editor is that it can be run off-site using a local disc filing system (DFS or ADFS), thus reducing the risks of security breaches. \***CONVERT** is run (on the network), T.GENFILE and T.PMAP are copied onto local disc: the network copies should then be deleted. All the edits (i.e. preparing mod-files and running \***MERGE**) can then be done whilst the computer is disconnected from the network. T.GENFILE, !mkdir and !rmdir are copied back onto the fileserver, and \***GENERATE** is run. T.GENFILE should then be deleted from the fileserver.

### General Suggestions

If the password file is fairly small then EDITPASS can be used. If an Archimedes is available then ARCPASS can be used (on virtually all sizes of password file). If the file is too big for EDITPASS then QEDIT can be used, subject to the limitations of QEDIT itself.

If a large number of users are being added or modified, then, whether the password file is large or small, we recommend that you use the batch mode editor. For extra security the batch mode system should be used off-site.

## Temporary files created by the batch mode editor

There are a number of temporary files used by the batch mode editor which are all held in the directory T. T.GENFILE and T.INT1 should be deleted (for security reasons) after a session has finished. The files are:-

T.GENFILE    T.PMAP    T.INT1    T.PASSTXT

T.INT1 and T.PASSTXT are temporary files created and used only by \*MERGE. The latter is the updated version of T.GENFILE and is normally \*RENAMED as this before MERGE exits. However, if MERGE fails it is possible that both T.GENFILE and T.PASSTXT will remain. Thus T.PASSTXT may be deleted at any time (except while MERGE is actually running).

There also are two files created by \*MERGE that will require be to \*EXECed by the user, which are:-

!mkdir        !remdir

### 5.5.1.1 Memory Requirements

\*MERGE requires HIMEM at &7C00 or greater. On a BBC microcomputer without shadow RAM, MODE 7 is required (and will automatically be selected if this is not already the case). If HIMEM is less than &7C00 and you have shadow RAM, MODE 131 will automatically be selected.

N.B. If HIMEM is less than &7C00 and you load \*MERGE, you will see the program being loaded into the screen. Normally, this does not matter because the first thing that the program does is to change to a different mode. However, if you in addition have \*OPT 1 1 set, there is a fair chance that the text printed by the OS will actually overwrite the loaded program, which will then crash.

You are therefore warned against using \*OPT 1,1.

If you are using a RISC OS computer then the \*CONVERT, \*MERGE, \*GENERATE suite of programs can be run using the 65Tube module.

### 5.5.2 \*CONVERT: Converting the existing password file

Syntax: \*CONVERT [<Disc name>]  
System Privilege Required.

Using the program CONVERT, a (human-readable) text file T.GENFILE is created from the current password file, %PASSWORDS. The discname is recorded in this file. In addition the program will create the file T.PMAP which contains a *bit-map* of all the personal account numbers currently allocated in the password file.

Note that users are allowed to modify some aspects of the password file themselves (by using \*PASS or \*OPT 4,n), so you should not use the old copy of T.GENFILE but create an up-to-date copy (you *cannot* use the 'last update date/time' to see whether the file has been directly modified in this way). However, if you wish to use a sequence of \*MERGE operations you must only run CONVERT once (if you run CONVERT immediately after MERGE, the file T.GENFILE will be overwritten and any modifications will be lost).

Having typed \*CONVERT the software will respond with :-

```
*Convert
Version 1.12, (C) SJ Research
Converts %PASSWORDS into text file T.GenFile
also makes T.pmap
```

For every ten users processed a dot will be printed.

If you want to look at the resulting file, type `*TYPE T.GENFILE`. To convert a password file other than the one on your currently selected disc specify the disc name after the `*CONVERT` command.

e.g. `*CONVERT MAIN`

### Corrupt %PASSWORDS files

CONVERT will give a warning when it finds corrupt URD/LIB pointers (i.e. those that point off the end of the file; pointers that point to other random places in the file could produce warnings of the URD/LIB text exceeding 80 chars).

## 5.5.3 \*MERGE and the mod-file

Syntax: `*MERGE [<mod-file name>]`  
System Privilege Not Required.

<mod-file name>, if not specified, defaults to `MODE`.

Changes to the password file are made by creating a *mod-file*. This file should contain a *mode* keyword, telling MERGE whether to add new users, modify existing users or remove existing users. There is then a section defining attributes that should apply to the new users: these are called *global* assignments. When removing users, this section is obviously not required. Then follows the list of usernames on which we wish to act. Each username is followed by a section (enclosed in curly brackets { } ) that defines actions to be done to that user only (these are called *local* assignments).

### Creating a Mod-file

A standard ASCII text editor is required. We suggest using Acorn EDIT (supplied with a BBC Master microcomputer), or WORDWISE on a BBC micro. VIEW can be used, but the format and justify options should be turned off and you should do not create any new rulers nor enter any formatting commands (shift-f8). EDWORD files are not suitable, but spooled output from this editor is acceptable. For very short mod-files it would be possible to use `*BUILD`.

The file may have any name as MERGE takes the filename as a parameter. Typically you might have a mod-file for each class held permanently on the fileserv, so that you can make changes to an entire class (e.g. remove them when they leave) very easily.

There may only be one occurrence of any given username in the mod-file.

*Global* keywords are specified outside a user definition and take effect for all the following users up to the next mode keyword. There may be many global assignments following each mode keyword; the assignments to a particular keyword are not cumulative (e.g. `ACC="1"`; followed later by `ACC="2"` is not the same as putting `ACC="1 2"`). All global keywords are reset by each mode keyword.

*Local* keywords are specified after a username within curly brackets { } and only affect that username. Local keywords take precedence over global keywords and again are not cumulative (e.g. global `Flag="Pw"`; followed by local `Flag="Ns"`; does not give `Flag="PwNs"`).

Comments can be specified by inserting an `"&"` as the first character of a line; the rest of the line up to a CR (CHR\$13) or a LF (CHR\$10) is then ignored.

There is one important restriction on the size of the Mod-file, that is that it cannot contain more than 256 users. However, this should not present a problem as MERGE can be run as many times as necessary on different mod-files, without having to re-run CONVERT or GENERATE.

The general form of a mod-file is shown below :-

.mode.

```
Global assignments
Username { local assignments }
Username { local assignments }
Global assignments
Username
```

```
.mode.
Global assignments
Username { local assignments }
.END.
```

## Running \*MERGE

```
*MERGE
*Merge [<mod-file>]
Version 1.16, (C) SJ Research
Parses mod-file and produces T.Int1,
then Merges T.Int1 with T.Genfile
Also produces !makdir & !remdir
```

```
Parsing Mod-file....
Warning - !makdir already exists.
Warning - !remdir already exists.
(O)verwrite, (A)ppend or (Q)uit
O/A/Q ?O
Merging T.INT1 & T.GenFile.
***** Error : at line 1
TONY {}
-----
SYST - User not found
...
Errors during Merge - aborted
```

## Errors

During the *parse* stage, the line number of the line in error and a relevant portion of the mod-file will be printed. During the merge stage, the line number and relevant portion of T.GENFILE will be printed. Lines are numbered starting from 1.

The following keywords may be defined either globally or locally :-

ACC, BASE, BOOT, CREDIT, FLAG, LIB, PACC, PASS, URD.

The keyword DEFAULT may only be defined locally.

In addition there are the following modes.

.ADD. , .REMOVE. , .MODIFY. , .END.

Please note that changing mode sets all global assignments to their defaults.

### 5.5.3.1 !makdir and !remdir

N.B. To use either of these files, the user requires system privilege, key in the SYSTEM position, and ownership of all accounts.

MERGE always creates, in the currently selected directory, the files !makdir and !remdir. These files contain a sequence of commands which will make/remove directory structures and also to credit/debit accounts, for any users added or removed during the merge process. In .ADD. mode, commands to create

the relevant URD directory and to credit the relevant account are added to !makdir, in .REMOVE. mode commands to delete the users' URD and all its contents and to debit the account are added to !remdir. In .MODIFY. mode, no commands are added to either file. To use the files, type

\*EXEC !makdir or \*EXEC !remdir.

N.B. Once either file has been \*EXEC'ed, it should be deleted by using \*DELETE !makdir or \*DELETE !remdir.

MERGE can also append new information to the end of an existing file, so that a sequence of MERGEs can be done, followed by a single \*EXEC command.

The following !makdir file was created as a result of the mod-file in §5.5.6. The disc name is Work. Bold type indicates user input.

```
*EXEC !makdir
*DIR :Work
*CDIR CLASS87
*DIR CLASS87
*CDIR FRED
*DEBIT 145 65535
*CREDIT 145 100
*ACCOUNT FRED 145
*DIR :Work
*CDIR TONY
*DEBIT 10B 65535
*CREDIT 10B 50
*ACCOUNT TONY 10B
*DELETE !makdir
```

The \*CDIR commands are inserted unless the URD keyword was defined to be the root, i.e. "\$" or ";<discname>". If URD=""; the directory \$.<username> will be created. The \*DEBIT, \*CREDIT and \*ACCOUNT commands are included whenever the user has a Personal account number (i.e. PACC<>").

An example of the contents of a !remdir file:-

```
*BASIC
LOAD"ERAQ"
RUN
:Work.CLASS87.FRED
N
*DEBIT 145 65535
```

For !remdir, URD and PACC definitions are taken from T.GENFILE. The "ERAQ" sequence of commands (lines 3 to 5) is present unless the URD is defined to be the root of any disc, the \*DEBIT is inserted when the user has a PACC. If some of the commands in !remdir are undesirable, they may be removed/modified with a text editor.

## 5.5.4 \*GENERATE

Syntax: \*GENERATE  
System Privilege Required.

The key need only be in the SYST position during the actual installation of %PASSWORDS, i.e. only for the very last phase of the program. Ownership of the root on the relevant disc is also required (usually account 0).

The GENERATE program creates a new filesaver-format password file PASSWDS from T.GENFILE.

GENERATE then installs this as %PASSWORDS and the user should then delete T.GENFILE and turn the key back into the SECURE position.

The file PASSWDS is always created in the root of the disc onto which the new password file needs to be written. This is because the file PASSWDS is transferred to %PASSWORDS using a \*RENAME command. This has the advantage of being an 'atomic' operation, i.e. no operations from other users are allowed while it is happening (especially logging-on). It also automatically deletes the original file. [This is a special case whereby \*RENAME is able to rename a file 'on top of' another already existing file; this does not work in reverse, and you cannot \*RENAME the password-file back out again.]

**Note:** Whenever the message Output now in file \$<discname>.PASSWDS occurs, this file will exist (with access "PWR/") and contain sensitive information until it is successfully installed as %PASSWORDS. Therefore, if the file is not installed, you should delete it. The file will also remain if you abort the installation.

GENERATE will only ask you whether you wish to install the new file when no serious errors were detected, and that the contents of the file are only guaranteed to be valid when there are no errors/warnings given. When GENERATE prompts the user, pressing any key other than 'Y' or 'y' will abort. Note that aborting in this way will leave the file in the root.

Running \*GENERATE:-

#### **\*GENERATE**

Password file editing system - GENERATE  
Reads file T.GENFILE, writes file \$.PASSWDS  
Version 1.08

.END. found

Number of users =&0022

Finished :  
00 warning(s)  
00 serious error(s)

Output now in file :SMALL1.PASSWDS

Install new file as %PASSWORDS ? (Y/N) Y

Installed.

## **5.5.5 Keywords**

All keyword assignments have the form:

<Keyword>="<value>";

Note the double-quotes and semicolon, which must always be present.

**ACC** Defines the group account numbers given to a user (see also PACC). Each account number is a number (in hexadecimal i.e. 0 to 9 then A,B,C,D,E,F) in the range 0 to 7FF. Multiple account numbers are separated by a space. To specify a range of accounts the first account number is the lower range followed by the '>' character followed by the upper range of the account number.

Group account numbers in the range 100..7FF are actually allocated in blocks of 40(hex). That is, if any account from 100 to 13F is specified, all accounts 100..13F will be owned by the user.

In .MODIFY. mode, the characters '+' and '-' may be used in the definition. All account numbers following such a character add or remove accounts as appropriate. A single definition may include both characters, and they will be evaluated on a left-to-right basis. E.g. ACC="0>7F -10>6F + 30 32" would give be equivalent to ACC="0>0F 30 32 70>7F". A whole account block can be removed by specifying a single account; for instance, ACC="-100" would remove accounts 100..13F.

When used in .ADD. mode, '+' or '-' will cause the error Bad character. In .REMOVE. mode, whilst local assignments are parsed, no error will be caused.

For example

```
ACC="10 2C 30>FF 140";
```

will assign account numbers 10, 2C, 30 through FF as well as group accounts 140..17F.

ACC defaults to "" i.e. no accounts.

## BASE

This defines the base user root directory name to which the username is added. It will generally be defined in a Global assignment, although it can be defined in a local assignment. No "\$." prefix is required - see the URD keyword.

For example:

```
.ADD.
```

```
BASE="CLASS87";
```

```
TONY {}
```

will give TONY the user root directory "\$.CLASS87.TONY"

BASE will override a URD assignment if it is defined at a later stage. Default for BASE is "\$" (and overrides until a URD is defined).

## BOOT

This defines the boot option for a user. The number ranges from 0 to 3 and has the following meaning on the BBC computer :-

- 0 - No action
- 1 - \*LOAD !boot
- 2 - \*RUN !boot
- 3 - \*EXEC !boot

The default value of BOOT is 0.

## CREDIT

In .ADD. mode (and no other), this will set the amount of space, in K, that is CREDITed to the personal account number. The password file itself will not be affected by the value of this keyword, only the !mkdir file is affected.

The number is in decimal and in the range 0-65535. The default value of CREDIT is 100.

## DEFAULT

This takes either 0 or 1 as a parameter. "1" sets this user to be the default user, "0" means that the user should no longer be so. MERGE never produces any warnings about 'silly' uses of DEFAULT (e.g. using DEFAULT="0"; on a user who wasn't the default user anyway).

May only be defined as a local keyword, and defaults to 0.

## DISCNAME

This keyword is always present in T.GENFILE and never anywhere else. It gives the name of the disc from which the password file came, and is put there by \*CONVERT.

## FLAG

For this keyword the assignment of data is in the form of two letter combinations which are

as follows.

Pw password locked  
Sy system privilege  
Ns No short saves  
En Permanent \*ENABLE  
Nl Library only used for \*RUN commands  
Ro '\*RUN only' user

A user with this option enabled may use \*RUN and certain other \*commands. Also permitted are FScall #14 (Read disc info), FScall #16 (Read date/time), FScall #25 (Read FS version number), and FScall #65 (Read/Write misc info). All other commands will give the error message Who are you?

Al Auxiliary account locked

When this is set, the user is not allowed to change the auxiliary account of any file or directory under any circumstances.

X2 Reserved

See Editpass for more information (section 4.3, page 4-8)

In addition to this, in .MODIFY. mode either '+' or '-' may precede any flags to either add or remove options.

For example:

```
.MODIFY.  
TONY {FLAG="+SyEn-Pw"; }  
will take the existing flag options set for TONY and add the system privilege and permanent *ENABLE and remove the option for password locked. If neither '+' or '-' are used in .MODIFY. mode then the new assignment will override the old definition for the flag.
```

Default is Flag=""; i.e. Password/\*OPT4 not locked, Not System Privileged, Short Saves allowed, \*ENABLE required (for wild-card \*DELETE), Library used for all operations, Not 'Run Only', Auxiliary Account not locked.

**LIB** This sets the initial library directory for the user. Default is "", which means that the fileserver will select \$.LIBRARY on the lowest numbered disc (a hard disc drive, if you have one).

**PACC** This keyword defines a personal account number and is a hexadecimal number in the range of 1 to 7FF. If set to "" it means that no personal account number is allocated. When used in a local assignment that particular personal account number is given to the user. If the personal account number has already been allocated to another user (as a PACC) then a warning will be given.

In .ADD. mode, when PACC is used in a global assignment it assigns the next free account number greater than or equal to the one specified. That account is then marked as allocated so that the next user will get a different account number. The file T.Pmap contains a map of the currently allocated PACCs, and this file is read and updated when using this feature. To disable this feature, set PACC="" in another global assignment; in a local assignment you would PACC="344" to assign a specific account number.

For example:

```
.ADD.  
FRED { PACC="500"; }  
STU { }
```

TONY {}  
.END.

will allocate personal account 500 to FRED, personal account 100 to STU (assuming it has not already been allocated to some other user) and personal account 101 to TONY. If personal accounts 100, 101 and 103 have been allocated to some other users then STU would be assigned personal account 102 and TONY would be allocated personal account 104.

Default is "100", i.e. allocation will start from 100 (.ADD. mode only).

**PASS** This sets a user's password, the default password is "".

**URD** This sets the user's root directory, and overrides any BASE definition. The fileserv selects the URD relative to the root of the disc on which %PASSWORDS is. Therefore you do not need to prefix it by "\$." unless the directory required is on a different disc (when you should use :<discname>). By default, BASE is set to "\$." and the URD is undefined (that is, it is not referenced). If URD is set to "", the URD becomes the default, i.e. \$.<username>. To set the URD to null, use URD="\$".

## Mode Keywords

**.ADD.** In this mode the user entries are taken as new users. If a user of this name already exists an error is generated. A set of commands are placed in the file "!mkdir" to create the appropriate URD and credit the appropriate personal account (the system manager will \*EXEC !mkdir at a later stage). If MERGE is used repeatedly, new commands will be appended to the existing !mkdir file, and a warning will be given. Therefore, once !mkdir has been \*EXECed it should be deleted.

If a user has PACC set to "" then the account number of the URD created will be the account number of the parent directory (i.e. no \*ACCOUNT command will be placed in the !mkdir file). In this case it is possible that the user will not have owner access to his URD.

**.REMOVE.** The specified users are removed from the password file. Obviously no global assignments or local assignments are needed, however it is not an error for these to exist. This makes it possible to remove blocks of users and later restore them just by changing the mode keyword to .REMOVE. The curly brackets {} must be present after each user name, although there needn't be any text within them. However, the text inside curly brackets is parsed, so don't put garbage in there!

A set of instructions is placed in the file "!remdir" to delete the appropriate URD and its contents and also to debit all the space allocated to that account.

**.MODIFY.** The data in the user entries is used to modify the data already held in an existing entry. It is an error for the user not to already exist. To add or remove accounts or flags from a user entry the characters '+' or '-' may be used.

**.END.** This signifies the end of data in both the mod-file and the gen-file. The use of .END. is optional, but \*CONVERT always puts a .END. at the the end of the gen-file.

## 5.5.6 Mod-file Examples

Consider the following mod-file :-

.ADD.

```
ACC="1 23"; BASE="CLASS87";
FRED {PACC="145";}
TONY { URD=""; CREDIT="50";}
.END.
```

The mode keyword `.ADD.` specifies that the users are to be added to the password file.

The next line is a global assignment: the keyword `ACC` is assigned the values 1 and 23 and the keyword `BASE` is assigned the name `CLASS87`; as these appear outside a username definition they are global assignments.

The username `FRED` has a local assignment defining his Personal Account Number as 145, so he will have access to Accounts 1 & 23 (from the global assignment) and Account 145, and his default directory after logon will be `$.class87`. `TONY` has the keyword `URD` defined locally which overrides the global `BASE` assignment.

User `TONY` will have a `URD` of `$.TONY`, and will have a personal account number allocated (the lowest free one above 100). He will also have access to accounts 2 and 23, but will only have 50k of disc space allocated to his personal account.

The following mod-file has exactly the same effect as the previous example.

```
.ADD.
TONY { URD=""; ACC="1 23"; }
FRED { ACC="1 23"; URD="CLASS87.FRED"; PACC="145"; }
.END.
```

A typical application of the batch mode editor would be to add a new year's entry to the system.

Suppose we have a mod-file called `CLASS4A` thus:-

```
.ADD.

PACC="200";
BASE="Class4a";
CREDIT="50";
FLAG="PwNsAl";

ArdleighW {}
BassetF {}
MunroeM {}
BunterW {PACC="";}
WilliamJ {}
BottVE {}
KermitF {}
BigglesDSO {}

.END.
```

By typing `*CONVERT , *MERGE CLASS4A , *GENERATE` you will now have an updated password file installed. To create the necessary directories, type `*EXEC !mkdir`. That's all there is to it.

You would normally keep the file `CLASS4A` around; in order to delete the users, change the `.ADD.` keyword to `.REMOVE.` and repeat the process, only this time finish off with `*EXEC !rmdir`.

## 5.5.7 Warnings and Errors

Warnings and errors are accompanied by a portion of the relevant file and two inverse exclamation marks

(\*CONVERT/\*MERGE) or a left-pointing arrow (\*GENERATE) to indicate the approximate location of the error. N.B. in mode 7, these symbols appear as white squares (character 255).

#### **\*CONVERT**

URD/LIB pointer corrupt for user - <username>  
EOF (no terminating user entry, or password file corrupt)  
Bad disc name  
%PASSWORDS not found  
Directory called T not found  
\*\_\*-\*\_\* System Error : <OS error message>

#### **\*MERGE - Parse stage**

>256 users in mod-file  
.END. is a global keyword  
Bad character  
Can't find mod-file  
Can't find T.pmap  
Default is only valid as a local keyword  
End of file inside quoted string (or missing ")  
Expecting a "  
Expecting a }  
Expecting a number  
Flag not known  
Keyword/userid too long  
Mode keywords must be specified globally  
Mode not specified, using .modify. by default  
Need an = to assign value  
No more personal account numbers!  
Number too large  
Parameter too long  
Personal account number already allocated  
T.pmap has not been generated by \*CONVERT  
Text found after .END.  
Unknown keyword  
User already exists (two users of same name in the mod-file)

#### **\*MERGE - Merge stage**

Bad keyword in Genfile  
Flag not recognised  
No users found in Mod-file!  
Second number in range smaller than first  
T.Genfile not found  
<Username> - User already exists in password file (in .add. mode)  
<Username> - User not found (in .modify. or .remove. modes)  
Warning - !mkdir already exists  
Warning - !remdir already exists

## **\*GENERATE**

### **Fatal errors:**

File not found - T.GENFILE

### **Errors:**

Discname too long

Mismatched { } brackets

Larger number of users in pass 2 - output file useless

Two users with same name, or not in alphabetical order

Userid is missing/zero length

\*\_\*\_\*\_\* System error : <OS error text>

### **Warnings:**

Bad number

Bad operator - expecting "=" or "{"

Bad range (second number after range indicator wasn't a number)

URD text exceeds 80 characters

BASE text exceeds 80 characters

Boot option >3

Constructed URD exceeds 80 chars

DEFAULT cannot be used as a global keyword

DISCNAME must be global keyword

Keyword/userid too long

Missing " in assignment to keyword

More than one default user - this one ignored

Odd number of characters in FLAG text

Password exceeds 10 characters

Significant text after .END. - ignored

Smaller number of users in pass 2

Start of range bigger than end

Unrecognised flag name

Unrecognised keyword

## **5.5.8 File Specification**

All characters ASCII 0 through ASCII 31 are considered as a SPACE. Top bits are stripped. There is no case sensitivity, as every alpha-numeric is taken as upper case.

<file> ::= <gen-file> | <mod-file>

<gen-file> ::= DISCNAME="<discname>"; <pw data> .END.

<mod-file> ::= .<mode>. <pw data> [mod-file] [.End.]

<pw data> ::= [<userdata> | <global assignment> | <comment>] [<pw data>]

<mode> ::= Add | Modify | Remove

<comment> ::= & <text> <line terminator>

<line terminator> ::= <CHR\$13> | <CHR\$10>

<global assignment> ::= <global keyword> = "<keyword value>";

<userdata> ::= <UserID> { [<local assignment>] } [<userdata>]

<local assignment> ::= <local keyword> = "<keyword value>"; [<local assignment>]

<UserID> ::= [<alphanum>]  
 <keyword> ::= ACC | BASE | BOOT | CREDIT | FLAG | LIB | PACC | PASS | URD  
 <global keyword> ::= <keyword> | DISCNAME  
 <local keyword> ::= <keyword> | DEFAULT  
 <keyword value> ::= <acc> | <lib> | <pass> | <boot> | <urd> | <flag> | <pacc> | <default> | <base>  
 | ""  
 <acc> ::= <hex> | <hex> > <hex> | -<acc> | +<acc>  
 <lib> ::= <path>  
 <pass> ::= <alphanum>  
 <boot> ::= 0 | 1 | 2 | 3  
 <urd> ::= <path>  
 <pacc> ::= <hex>  
 <default> ::= 0 | 1  
 <base> ::= <path>  
 <flag> ::= [<flagsymbol> | +<flagsymbol> | -<flagsymbol>]  
 <flagsymbol> ::= Pw | Sy | Ns | En | NI | Ro | Al | X2  
 <path> ::= <name>[.<path>] | :<discname>[.<path>]  
 <discname> ::= <alphanum>  
 <name> ::= <alphanum>  
 <hex> ::= <hexit> | <hexit><hexit> | <hexit><hexit><hexit>  
 <hexit> ::= 0|1|2|3|4|5|6|7|8|9|A|B|C|D|E|F

## 5.5.9 Known Problems

### General:-

Whilst all programs set protection against remote network operations (\*VIEW etc) during operation, the current versions of CONVERT/MERGE do not clear RAM before exiting (GENERATE does), although they do leave the computer protected after exiting. Therefore, you should always logoff and then switch the computer off (the order is important) after you have finished using these programs.

### \*CONVERT, version 1.12:-

A corrupt %PASSWORDS file (with no terminating user entry) will give an EOF error. The resulting T.GENFILE will contain useful information, but T.PMAP will not have been saved. You should \*TYPE T.GENFILE to find out whether most of the users have been included; there may be some corrupt users at the end of the file - these should not matter. Then, using \*GENERATE (which does not require T.PMAP) you can create a repaired password file, which can then be re-CONVERTed correctly.

Some fatal errors (e.g. Account Bankrupt, Disc full, and Network errors) cause CONVERT to abort without closing files.

If the DEFAULT USER pointer is corrupt, CONVERT will not produce a warning; no user will have DEFAULT="1" in the gen-file.

### \*MERGE, version 1.16:-

\*MERGE does not give a warning when more than one user has been assigned DEFAULT="1". GENERATE will however give a warning, and will ignore subsequent assignments. In the case where DEFAULT="0" is specified when the particular user was not already the default user, again no warning will be given.

High-numbered group accounts do not act as blocks e.g. if you have ACC="100>17F" in T.GENFILE, the do ACC="-140", the result will be ACC="100>13F 141>200". GENERATE however does treat them in blocks, so will give access to A/cs 100>200 as before. Actually this is not normally a problem, since CONVERT does not produce ranges for high numbered accounts (ranges are only produced by MERGE), it merely specifies the base account number. From a password file it would have given ACC="100 140" whereupon removing account 140 using MERGE would have had the desired effect.

### \*GENERATE, version 1.08:-

Does not give a warning if there are no system privileged users with access to account 0.

## 6.3 Setting up the Printers

The File Server has to keep a number of pieces of information about the printers connected to it. The electrical parameters of the serial printer (baud rate etc.) are kept within the fileserver itself, but all the other information is kept on disc. Most of the settings are kept in a special part of the disc which can only be accessed by the **Editprint** program. If you are using more than one disc, this information will be kept on the first disc in the system (i.e. the first in the list produced by \*FREE). This will be a hard disc if you have one, but will otherwise be the floppy disc in drive A. If you use different floppy discs in this drive, remember that you will need to use **Editprint** on each of them. The **Editprint** information is copied by the *copy discs* option in utility mode.

Further information is kept in *banner files*, but these are ordinary files which require no special precautions.

### 6.3.1 Editprint

**Editprint** is a BASIC program, which can only be used by system privileged users. To adjust the printer settings, type:

**CHAIN "EDITPRINT"**

The program will respond with the editing screen similar to the one below :

- EDITPRINT - Printer Setup for SJ Research File Server 254					
Name	Status	Output to	Anonymous Printing	Account Required	AccNo Default
parall	Spooling	Parallel	No	No	
serial	Spooling	Serial	No	No	
hold	Hold		No	No	
auto	Auto	1st:parall 2nd:serial			
	Off				
	Off				
	Off				
	Off				
	Off				
	Off				
	Off				
	Off				
	Off				
	Off				
	Off				
	Off				

Esc: Quit                      Space: Toggle Data                      f3: Save Details & Exit

You can move round the screen using the cursor keys. Your current position in the screen is shown by displaying the field in reverse video. Editing a field can be done by pressing <Space> to rotate through the possible legal values. Alternatively pressing the initial letter of the value you want will enter that value.

Although there are only two physical printers attached to the File Server there are 16 logical printer names which can be used to access these printers. Each of these logical printers can have a different set of properties and these are displayed in the row across from the name.

**Name** is the name which users will quote to specify that particular logical printer. Printer names may be up to six characters long. The name **PRINT** is reserved and must not be given to a printer. If the printer name is blank (i.e. consists of spaces), that printer is disabled completely.

**Status** determines where output sent to this logical printer will be directed. There are four possible values.

**Off**                      This disables the printer completely so that it cannot be selected using either \*PS or

## \*PRINTER

Spooling	Printer output sent to a spooling printer is stored as a print job file in the %PRINTQ directory prior to being sent to the printer. For fast printers, it is preferable to spool to disc, preventing one user from claiming the printer for an extended period. For slow printers, or graphics dumps, it saves time to start printing immediately.
Non-spooling	Printer output sent to a non-spooling printer will be output immediately if the relevant physical printer is free. If the physical printer is busy then output will be spooled.
Hold	Printer output sent to a hold printer will be stored as a print job file in %PRINTQ however the File Server will not attempt to output the data stored to a physical printer. Hence it is possible to have two logical printers named 'PAPER' and 'LABELS', only one of which is enabled at any time. Users can generate both types of output, and any documents sent to the hold printer will be held until someone changes the stationary in the printer and uses EDITPRINT to enable the corresponding printer name. Another use for hold printer names is to allow users to generate output for a <i>remote despooler</i> program: this ensures that the File Server itself does not try to print jobs intended for a distant printer.
Auto	Printer output sent to an auto printer will go to one of two possible other logical printers, a first choice and a second choice. At the time of printing the File Server will attempt to allocate the first choice printer and if this is not possible will try to allocate the second choice printer. It is conventional to set up the <b>Auto</b> printer such that the first choice is the fastest printer for long listings, with the second choice being the other printer with a similar banner. If the second printer is unsuitable for listings, the auto printer would usually specify just one choice, or be turned off altogether.

**Output to** determines which physical printer will be used for output sent to a logical printer. In the case of an auto printer this column will hold the first and second choice logical printers.

**Anonymous Printing** controls whether users who have selected this logical printer, but are not logged on to the File Server, may print. If such a user presses <CTRL B> then he will be logged on as **ANONPRINT** or the default user if **ANONPRINT** does not exist. Having finished printing the user will be automatically logged off.

**Account Required** controls whether a user requires a specific account number to select this logical printer.

**AccNo** specifies the account needed if Account Required has been set to Yes.

**Default** specifies whether this is a default printer or not. If a user attempts to print without explicitly selecting a printer by name then the File Server will try to allocate a default printer. The File Server works down the list of printers defined as default printers and will allocate the first default printer which the user is allowed to use. Popular settings for the default printer are Auto or Hold.

Having edited details they are saved by pressing the **f<sub>0</sub>** key. Before saving some checking is performed to ensure that in particular there is no combination of auto printers which is circular. If this is found to be the case then an error is given and the cursor is move to the printer definition which is incorrect.

**Banner file** gives the name of a text file which controls the *banner* that is printed at the top of all printer output. The various possibilities for the contents of the banner file are described in section 6.3.2. The file name is looked up starting from \$ on the first disc drive, so **banners.fancy** would be equivalent to **\$.banners.fancy**. If the file cannot be found, or if the name is blank, no banner is printed at all: this is useful for non-standard devices such as graph plotters. Note that the system must have read access to the

banner file: the access string on the file would usually be set to **WR/**.

### 6.3.2 The Banner File

The printer server usually prints a heading at the start and end of each piece of printed output, known as the *banner*. The format of the banner is controlled by a file associated with each printer name, and it may contain both fixed text and some information about that job, such as the name of the user that generated it and the date. The file to be used is stored in the directory **\$.BANNERS** on the lowest numbered File Server disc and has the name as the logical printer name (see section 6.3.1). Eg a printer with a logical name of **parall** will require a banner file **\$.BANNERS.parall**. It is possible to have two or more logical printer names which specify the same physical printer but with banner files containing different data: a printer name **NLQ** might have a banner file containing the necessary control codes to set that printer into near letter quality mode, while there would be another printer name **DRAFT** which used the same printer but had a banner file containing the necessary control codes to set that printer into draft mode.

If the expected banner does not exist, or the system has not got read access to it (this means that the file must have letter **R** owner access), then the users' text will be printed without a banner or endtext: no error message will be produced.

The disc supplied with the File Server has a directory **\$.BANNERS**, containing (initially) three sample banner files, called **SIMPLE**, **FANCY** and **EpsonNLQ**. The other files in **\$.BANNERS** are **PARALL** and **SERIAL** for use with the printers set up as the File Server is supplied, and these are simply copies of the **SIMPLE** banner file. The system manager will need to create new banner files for each of the new printers created.

The banner file is a simple text file, of the sort created by **\*BUILD** or **Wordwise**. It contains a mixture of straightforward text, which is just printed out, and special symbols which are replaced by the information they represent. The file is split into two parts: the *banner* which is printed at the top of a user's output, and the *endtext* which is printed at the end, separated by the special symbol **<BANNER>**.

end-text<BANNER>banner text

The special symbol **<BANNER>** must be enclosed in angle brackets as shown. Note that the end-text (the characters to be printed after the user's output) comes first, and the banner text itself after the word **<BANNER>**. The texts will be printed literally until a special symbol is encountered.

The banner or end-text may contain special symbols from the list below. Note that all the symbols are enclosed in angle brackets **<>**. A carriage return character in the text will cause a carriage return on the printer; note that if your printer does not do an automatic line-feed after each carriage return, then a line feed character (or **|J**) should be put after each carriage return character (unless you intend to over-print lines).

There are three symbols that do not cause anything to be printed directly, but they select which of three possible times are printed when the identifiers **<HOURS>**, **<MINUTES>** etc. are used.

**<NOW>** selects the current time of day, at the moment when printing is actually taking place. The time to be printed is frozen at the instant when the **<NOW>** symbol is processed: this avoids inconsistent results if the clock ticks between the printing of the hours and minutes. If another **<NOW>** is encountered (in the end-text, for example), this will freeze a different value, as time will have elapsed during the printing of the intervening text.

**<START>** selects the time of day at which the printing job was initiated, from the user's computer. Note that, especially when print spooling is used, this (and **<END>** below) may be substantially earlier than the time given by **<NOW>**.

**<END>** selects the time of day at which the user finished sending characters to the printer.

The following identifiers cause part or all of the time and date as selected above, to be entered into the banner or end-text string. The default time selection is **<START>**.

```

460 LOCAL redo
470 redo=5
480
490 REPEAT
500 tx_blk%?0=&83
510 tx_blk%?1=&9C :REM Broadcast on port &9C
520 tx_blk%?2=&FFFF :REM Broadcast operation
530 $(tx_blk%+4)="BRIDGE"
540 tx_blk%?10=reply_port
550 tx_blk%?11=network%
560 X%=tx_blk%:Y%=tx_blk% DIV 256
570 A%=&10:CALL osword
580 A%=&32
590 REM Wait for completion
600 X%=FNpoll tx
610 redo=redo-1
620 UNTIL redo=0 OR X%=&40 OR X%=&43 OR X%=&44 OR X%=0
590 REM Wait for completion
600 X%=FNpoll tx
610 redo=redo-1
620 UNTIL redo=0 OR X%=&40 OR X%=&43 OR X%=&44 OR X%=0
630 IF X%>0 THEN PRINT"Broadcast failed, error "&~X%:PROCdelete:END
640ENDPROC
650
660DEF PROCdelete
670 REM delete a receive block
680 A%=&34
690 X%=blk%?0
700 CALL osbyte
710ENDPROC
720
730DEF FNpoll tx
740 REPEAT UNTIL NOT ((USRosbyte) AND &8000)
750=(((USRosbyte) AND &FF00) DIV &100)
760
770DEF PROCdisplay(no%)
780 PRINT"Found network ";no%
790 PROCdelete
800 PROCset_up_rx
810ENDPROC

```

## 10.19 Printers

The protocol used by SJ Research to locate named printer resources is an adaptation of the protocol used by Acorn (and implemented in ANFS). The extra complexity is introduced by the possibility of having multiple printers at the same station, such that the printer server has to keep track of the current printer selection for each user. The printer status protocol is used : a status request is broadcast, containing the name of the printer desired, and the station, making the broadcast can chose which of the responding servers to select.

### Print Server Status Enquiry Protocol

#### Client to Print Server

Port	&9F
Data-Data+5	6 character printer name padded with spaces
Date+6-Data+7	Two byte reason code (=1 for print status enquiry, =6 for printer name)

#### Print Server to Client

Port	&9E
Data	1 byte status report
Data+1-Data+2	Two byte station number of machine with which the server is busy

Bits 0-2 of the status byte give the status of the client's input to the printer via the network. Bits 3-4 give the status of the output from the print server to the printer. Bits 5-7 are reserved for future use and currently return zero. Currently defined status values are :-

#### Input

- 0-Ready
- 1-Busy
- 2-Jammed (general software problem)
- 3-Jammed, due to printer offline (general hardware problem)
- 4-Jammed, due to disc full, directory full or similar
- 5-User not authorised to use printer
- 6-Spooler going offline / operator has barred input
- 7-Reserved

#### Output

- 0-ready
- 1-Printer offline
- 2-Printer jammed (ie has not accepted data for a long time)

Typical responses include :

- 00000-Normal OK status
- 01000-Printer offline, but spooler still accepting data
- 00100-Printer OK, but disc temporarily full
- 01100-Printer offline, disc full

Print drivers should poll the printer as usual and proceed only if (status AND 7)=0

## Read/Write System Info.

Function code=65

### General description

This call provides an interface to read and write the printer information and change the privilege needed to write the file server's time. All of these read calls are unprivileged commands. All the write operations are privileged.

### Reset print server information :

This call is used to reset the printer information and **must** be issued for any of the other change printer information calls to take effect.

#### On entry,

0	Transmit block (shown on summary page)
7	
8	ARG=0

#### On exit,

0	Receive block (shown on summary page)
4	

### Read current state of printer :

This call returns the detailed information about a logical printer.

0	Transmit block (shown on summary page)
7	
8	ARG=1
9	Printer number (1 - 8)

#### On exit,

0	Receive block (shown on summary page)
4	
10	Name of printer (padded with spaces)
11	Bit 3 - Spool to disc Bit 2 - Account ownership required Bit 1 - Anonymous users allowed Bit 0 - Printing enabled
13	Account number (Only relevant if bit 2 of previous byte is set)
n	Banner file name (terminated by a CR if less than 23 characters)

### Write current state of printer

This call writes the detailed information about a printer, system privilege is required to do this. This call is not supported on SJ Fileservers version 1.00 or greater; see ARG=16 for call to write printer information.

#### On entry,

0	Transmit block (shown on summary page)
7	
8	ARG=2
9	Printer number (1 - 8)
15	Name of printer (padded with spaces)
16	Bit 3 - Disable spooling to disc Bit 2 - Account ownership required Bit 1 - Anonymous users allowed Bit 0 - Printing enabled
18	Account number (Only relevant if bit 2 of previous byte is set)
n	Banner file name (terminated by a CR if less than 23 characters)

#### On exit,

0	Receive block (shown on summary page)
4	

### Read the AUTO printer priority

This call reads the order in which printers are selected for users who have not requested a particular printer. This call is not supported on SJ Fileservers version 1.00 or greater.

#### On entry,

0	Transmit block (shown on summary page)
7	
8	ARG=3

#### On exit,

0	Receive block (shown on summary page)
4	
	Number of printer entries available (current implementation=2)
5	
	1st choice of printer
6	
	2nd choice of printer
7	
	.
	.
	.
n	

### Write the AUTO printer priority

This call allows a privileged user to write the order in which printer are selected for users who have requested the AUTO printer. This call is not supported on SJ Fileservers version 1.00 or greater.

#### On entry,

0	Transmit block (shown on summary page)
7	
8	ARG=4
9	Default printer 1
10	Default printer 2
	.
	.
(n+8)	
(n+9)	Default printer n

#### On exit,

0	Receive block (shown on summary page)
4	

### Read system message channel

This call returns the physical printer that all system messages are sent to. Note that the printer is a physical printer, so the parameter should be either 1 (parallel) or 2 (serial).

#### On entry,

0	Transmit block (shown on summary page)
7	
8	ARG=5

#### On exit,

0	Receive block (shown on summary page)
4	
5	Current system message printer

## Write system message channel

This call allows a privileged user to set the physical printer that system messages come out of.

### On entry,

0	Transmit block (shown on summary page)
7	
8	ARG=6
9	New system message printer

### On exit,

0	Receive block (shown on summary page)
4	

## Read message level

This call reads the current level of system messages. The value returned is in the range of 0 to 255. The amount of output is the level of output selected plus all the levels below that level. Therefore, in the list of levels shown to set the message level to 7 would make the file server print all logons and logoffs as well as errors.

Message level	Description
0	Off
5	Logon/logoff
7	Errors (i.e. 'Wrong password', 'bad name' etc.)
10	Maximum users and all star commands
11	Load/save
15	*cat and opens
128	Aborted loads
130	Function codes
150	Network errors
170	Map building names
200	Disc read/write
250	All successful network transactions to and from the fileserv
255	All Activity to the JPROC processor

### On entry,

0	Transmit block (shown on summary page)
7	
8	ARG=7

### On exit,

0	Receive block (shown on summary page)
4	
5	Current message level

## Set message level

This call sets the message level, as described above. It should never be necessary to set the message level to greater than 127 and that setting the message level to a value greater than 150 produces excessive output and will probably reduce the performance of the file server.

### On entry,

0	Transmit block (shown on summary page)
7	
8	ARG=8
9	New message level

### On exit,

0	Receive block (shown on summary page)
4	

## Read the default printer

This call returns the default printer. This printer will be selected if a user starts to print without having selected a particular printer. This call is not supported on SJ Fileservers version 1.00 or greater.

The values for the default printer are :-

- 0 Automatic search through the list default printer priority list (set by Fn=65 ARG=4)
- 1 Logical printer 1
- 2 Logical printer 2
- .
- .
- 8 Logical printer 8

255 Hold the job output in the %PRINTQ directory.

### On entry,

0	Transmit block (shown on summary page)
7	
8	ARG=9

### On exit,

0	Receive block (shown on summary page)
4	
5	Current default printer

## Set the default printer

This privileged call sets the default printer, see ARG=9 for more information about valid printer numbers.

### On entry,

0	Transmit block (shown on summary page)
7	
8	ARG=10
9	New default printer setting

### On exit,

0	Receive block (shown on summary page)
4	

## Read the privilege required to change time

This call reads the privilege required to change the file servers time. The SJ Research file server normally insists on system privilege to be able to change the time. However, if it is desired to change the time frequently this can be disabled.

### On entry,

0	Transmit block (shown on summary page)
7	
8	ARG=11

### On exit,

0	Receive block (shown on summary page)
4	
5	0 - Privilege required 1 - Privilege not required

## Set the privilege required to change time

This call sets the privilege required to change the file servers time.

### On entry,

0	Transmit block (shown on summary page)
7	
8	ARG=12
9	0 - Privilege required 1 - Privilege not required

### On exit,

0	Receive block (shown on summary page)
4	

## Read printer information

This call is only available on SJ File Servers version 1.00 or greater. This one call replaces the several calls previously needed to read all printer information.

### On entry,

0	Transmit block (shown on summary page)
7	ARG=15
8	Number of printers on which to return information
9	Start printer number (0-15)
10	

### On exit,

0	Receive block (shown on summary page)
4	Number of printer entries returned
5	Printer name
11	Status of printer 0-Off 1-Spooling 2-Non-spooling 3-Hold 4-Auto
12	1-Default printer 0-Not a default printer
13	1-Anonymous users allowed 0-No anonymous users allowed
14	1-Account number required 0-No account number required
15	Account number (Only relevant if previous byte was 1)
17	Output port 1-Parallel 2-Serial or 1st choice printer no. for Auto
18	2nd choice printer no. for Auto
19	Reserved
21	Next printer (as above bytes, 5-20)
n	

## Write printer information

This call is only available on SJ File Servers version 1.00 or greater. This one call replaces the several calls previously needed write all printer information.

### On entry,

0	Transmit block (shown on summary page)
7	ARG=16
8	Number of printers for which to write information
9	Start printer number (0-15)
10	Printer name
16	Status of printer 0-Off 1-Spooling 2-Non-spooling 3-Hold 4-Auto
17	1-Default printer 0-Not a default printer
18	1-Anonymous users allowed 0-No anonymous users allowed
19	1-Account number required 0-No account number required
20	Account number (Only relevant if previous byte was 1)
22	Output port 1-Parallel 2-Serial or 1st choice printer no. for Auto
23	2nd choice printer no. for Auto
24	Reserved
26	Next printer (as above bytes, 10-25)
n	

### On exit,

0	Receive block (shown on summary page)
4	

## Read Encryption Key

### General description

This call supplies an encryption key.

Function code=66

## Read/Write Backup

Function code=67

### General description

This call provides an interface to read and write information necessary for an automatic backup. The call can also be used to read the tape id block. If there is no error the backup is possible.

### Determine whether backup is possible

This call is to determine whether a tape backup is currently possible.

#### On entry,

0	Transmit block (shown on summary page)
7	
8	ARG=0

#### On exit,

0	Receive block (shown on summary page)
4	

### Read tape id block

This call reads data from the tape Id block of the tape currently loaded.

#### On entry,

0	Transmit block (shown on summary page)
7	ARG=1
8	offset from start of block
10	number of bytes to return (should be <128)
12	

#### On exit,

0	Receive block (shown on summary page)
4	
n	Data from tape Id block

## Read current status of auto backup

This call returns the details of any auto backup currently pending.

#### On entry,

0	Transmit block (shown on summary page)
7	
8	ARG=2

#### On exit,

0	Receive block (shown on summary page)
4	
5	0-No backup pending 1-Backup pending
10	Time of backup in standard format Byte 5 - Day Byte 6 (Top 4 bits) - Years since 1981 Byte 6 (Bottom 4 bits) - Month Byte 7 - Hour Byte 8 - Minutes Byte 9 - Seconds
11	Printer output 0 - Off 1 - Parallel 2 - Serial
12	Tape partition for backup currently set to 0
22	Disc name to backup (terminated by a CR if less than 10 characters)
32	Tape name to use for backup (terminated by a CR if less than 10 characters) A null name means any tape can be used

## Write current status of auto backup

This call is used to set an auto backup or to cancel a previously defined auto backup.

### On entry,

0	Transmit block (shown on summary page)
7	ARG=3
8	0-Cancel backup 1-Set a new backup (Following bytes are only relevant if setting a backup)
9	Time of backup in standard format Byte 9 - Day Byte 10 (Top 4 bits) - Years since 1981 Byte 10 (Bottom 4 bits) - Month Byte 11 - Hour Byte 12 - Minutes Byte 13 - Seconds
14	Printer output 0 - Off 1 - Parallel 2 - Serial
15	Tape partition for backup currently set to 0
16	Disc name to backup (terminated by a CR if less than 10 characters)
26	Tape name to use for backup (terminated by a CR if less than 10 characters) A null name means any tape can be used
36	

### On exit,

0	Receive block (shown on summary page)
4	

N.B. These calls are only supported on the SJ Research file server.

## 10.21 Tape Id Block Format

0	"SJ Research"
11	0=blank 1=used
12	Tape name (terminated with CR if less than characters)
22	Number of passes
24	ASCII description terminated with CR
104	Date and time when formatted Byte 104 - Days Byte 105 (Top 4 bits) - Year since 1981 Byte 105 (Bottom 4 bits) - Month Byte 106 - Hours Byte 107 - Minutes Byte 108 - Seconds
109	
128	Reserved

There then follow 1-14 entries of the form

0	Low 2 bits: 0-Blank 1-OK 2-Corrupt Bit 7 Set if non-MDFS disc
1	Disc name backed up
11	Date and time of backup Byte 11 - Days Byte 12 (Top 4 bits) - Year since 1981 Byte 12 (Bottom 4 bits) - Month Byte 13 - Hours Byte 14 - Minutes Byte 15 - Seconds
16	Data start block
19	Length in K
22	Error info
23	
64	Reserved

## 10.22 Password File Format

The password file, and the user information stored by the fileserver, is stored in the file %PASSWORDS. The following information is included for users who wish to write their own password handling programs eg password disclosing. It follows the following format :-

0	Entry number of 1st user entry with the first character less than 'A'
2	Entry number of 1st user entry with the first character as 'A' or 'B'
4	Entry number of 1st user entry with the first character as 'C' or 'D'
6	.
28	Entry number of 1st user entry with the first character greater than 'Z'
30	Entry number of the default user
32	Not used
64	1st User entry
128	2nd User entry
192	3rd User entry
256	.
n	Terminating User entry (Filled with &FF)
(n+64)	URD names and Libraries (pointed to by the user entry) Each a maximum of 80 characters terminated by a <CR>.
z	

Each user entry (occurring in the password file every 64 bytes) has the following format :-

0	User Identifier (Terminated by a <CR> if less than 10 characters) Top bit must be masked out using AND &7F
10	Password (Terminated by a <CR> if less than 10 characters) Top bit must be masked out using AND &7F
20	Boot option
21	Flag bit 0 = Password unlocked bit 1 = System privileged bit 2 = No short SAVEDs bit 3 = Permanent *ENABLE bit 4 = Acorn style Library bit 5 = Run only user bit 6 = Can't change auxiliary account
22	Offset from start of file to user's Root Directory Set to 0 if URD is normal
25	Offset from start of file to user's Lib. Directory Set to 0 if LIB is normal
28	Personal account number (0 = no personal account)
30	Bit map of high numbered accounts bit 0, &600-&63F bit 7, &7C0-&7FF
31	Reserved
32	Bit map of account ownership bit 0, byte 0=Account 0 bit 7, byte 1F=Account 255
64	

The top bits of the User Identifier and the Password hold a bit map of ownership to high numbered accounts.

bit 0, byte 0=Accounts &100-&13F  
bit 7, byte 19=Accounts &5C0-&5FF

Setting the 'run only user' bit limits that user to calls 5,14,16,23,25,65, and 66, and commands \*SDISC, \*DIR, \*LIB, \*BYE and \*I AM.

This information is not available on Acorn File Servers.